Walter Carnielli
Marcelo Coniglio
Dov M. Gabbay
Paula Gouveia
Cristina Sernadas

Applied Logic Series 35

Analysis and Synthesis of Logics

How to Cut and Paste Reasoning Systems

Springer

Analysis and Synthesis of Logics

APPLIED LOGIC SERIES

VOLUME 35

SCOPE OF THE SERIES

Logic is applied in an increasingly wide variety of disciplines, from the traditional subjects of philosophy and mathematics to the more recent disciplines of cognitive science, computer science, artificial intelligence, and linguistics, leading to new vigor in this ancient subject. Springer, through its Applied Logic Series, seeks to provide a home for outstanding books and research monographs in applied logic, and in doing so demonstrates the underlying unity and applicability of logic.

The titles published in this series are listed at the end of this volume.

Analysis and Synthesis of Logics

How to Cut and Paste Reasoning Systems

by

Walter Carnielli
University of Campinas (UNICAMP), Brazil

Marcelo Coniglio
University of Campinas (UNICAMP), Brazil

Dov M. Gabbay
King's College, London, United Kingdom

Paula Gouveia
Instituto Superior Tecnico, Lisbon, Portugal

and

Cristina Sernadas
Instituto Superior Tecnico, Lisbon, Portugal

Springer

Library of Congress Control Number: 2008920294

ISBN 978-1-4020-6781-5 (HB)
ISBN 978-1-4020-6782-2 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

Contents

Preface

The aim of this book is to show how logics can be cut and paste in order to be applied to express and model problems in several distinct areas. The universal applicability of logic in both pure and applied science is a fact that defies philosophers. Contemporary logical research, however, has an undeniable tendency towards pluralism and compartimentation, as shown by the division of philosophical logic in areas and subfields. On the one hand, we have logics alternative to classic, such as many-valued logic, intuitionistic logic, paraconsistent logic. On the other hand, we also have logics complementary to classic, such as modal logics, and, in particular, temporal logic, epistemic logic, doxastic logic, erotetic logic, deontic logic, and so on.

Considering that reasoning is through process, this compartimentation, even if driven by methodological and technical reasons, has been said to be harmful to logic while a philosophical discipline. From this viewpoint, combinations of logics goes in the opposite direction of restoring the entirety of logic as wide theory of rationality, much in the same spirit to what happens in areas as algebraic geometry. Thus, from a philosophical perspective, logical combinations of tense and modality, for instance, may offer a better look to issues in the theory of causation and action. Combining temporal logic with alethic modal logics adds a temporal dimension to knowledge and belief.

Conceptually, the idea of looking to logic as an entirety instead of isolated fragments is not new. Philosophers and logicians from Ramón Lull (1235–1316), in his *Ars Magna*, to Gottfried W. Leibniz (1646–1716), with his *calculus ratiocinator* [179], have dreamed of building schemes or even machines that can reason by combining different logics or logic-like mechanisms that could cooperate instead of competing.

The activity of combining logics, as seen nowadays, offers an important tool for modularity. Rather than building a logic from scratch, it may be better for some applications to depend upon previous work on specialized topics. The underlying idea is that logics can be reusable, leading to a perspective gain with the resulting combined system. However, there are many technical difficulties if one is interested in the practical activity of combining logics. Symbols may mean different things in different logics. How is it possible to define the languages in order to compose them into an organic entity? Also, proofs and derivations can have different meanings in different logics. How to thread rules and derivation schemes of totally different nature?

Combining logics have also a surprising impact on philosophical questioning, an aspect that this book is not aimed to, but that should not be overlooked. An illustrative example is the well-known David Hume's objection from concluding a normative statement of the form "ought to be" starting from a descriptive statement of the form "what is" (a much discussed question in moral theory). So, for instance, from statements of fact, such as "emission of carbon dioxide is harmful to society", a statement of obligation such as "all nations ought to follow mandatory emission limitations" could not, according to an interpretation of Hume's ideas, be derived.

From the point of view of combining logics, this question is strictly connected to accepting properties of combining deontic and alethic logics, such as $p \rightarrow (Op)$, where O is the deontic obligation operator and p is an assertion. Such formula is what we call a "bridge principle". The term, meaning specifically a statement that binds factualities to norms, appears already in [3] and subsequently in [262].

In our treatment, by "bridge principles" we mean, in a wide sense, any new derivations among distinct logic operators (new in the sense of not being instances of valid derivations in the individual logics being combined).

Another much discussed thesis is the famous "ought-implies-can" thesis attributed to I. Kant, according to which the fact that we ought to do something implies that it has to be logically possible to do. This would be formalized through the following bridge principle: $(Op) \rightarrow (\Diamond p)$, where the diamond $\Diamond$ denotes the alethic "possibility" operator. Thus the principle means that if an assertion is obligatory then it must be possible. Other interpretations suggest that what Kant believed is that we cannot be obliged to do something if we are not capable of acting in that way. This would be formalized by the (non necessarily equivalent) bridge principle $(\neg\Diamond p) \rightarrow (\neg Op)$.

Not only bridge principles have an underlying conceptual meaning, but they also may emerge spontaneously, with surprising consequences as we show in many places of this book. The influence of bridge principles is yet perceived in the way the *collapsing problem* (a phenomenon of combining logics by which, for instance, intuitionistic collapses to classical logic) is solved (see Chapter 8).

This book intends to address the questions presented above in detail, presenting with the foremost rigor the issues of logical manipulation. The reader will learn here how to set up the syntactical dimension in detail, and how to define the semantics and the proof theory for recombinant logics. The impact of combination of logics in practice can only be assessed by people involved in the application domains. However, we believe that these techniques can be useful in fields such as computational linguistics, automated theorem proving, complexity and artificial intelligence. Other promising applications are in the areas of software specification, knowledge representation, architectures for intelligent computing and quantum computing, security protocols and authentication, secure computation and zero-knowledge proof systems and in the formal ethics of cryptographic protocols.

Combinations of temporal reasoning, reasoning in description logic, reasoning about space and distance are becoming a relevant toolbox in modeling multi-agent systems. The resulting hybrid systems have the main advantage of combining logics which would be otherwise incompatible. Proof procedures with controlled complexity, model checking and satisfiability checking procedures can be obtained for a bigger logic from the respective procedures for the component logics.

But the reader should not think that combinations of logics is a topic restricted to applications outside logic. On the contrary, although we do not deal with this question in this book, the very idea of combining logics, as we see it, touches on more abstract domains as applied to the logical theory itself: for instance, as suggested in [233]. The idea of combining logics can be even useful to understand apparently far away topics such as Popper's structuralist theory of logic, as in [223], where an elementary theory of combining negations was developed.

In a rigorous way, the problem of combining logics can be seen as follows: given two logics $\mathcal{L}_1$ and $\mathcal{L}_2$ we want to combine them and obtain a new logic $\mathcal{L}$ satisfying certain requirements. In general, there are several mechanisms to combine the original logics. Choosing mechanism $\odot$, the new logic is $\mathcal{L} = \mathcal{L}_1 \odot \mathcal{L}_2$. That is, $\odot$ is an operator on some class of logics including $\mathcal{L}_1$ and $\mathcal{L}_2$. Different operators may lead to different resulting logics. Most of the operators provide an algorithmic construction of logic $\mathcal{L}$ by stating its language, semantic structures and/or deductive systems. Moreover, the construction of $\mathcal{L}$ usually is a minimal (or maximal) construction. The combined logic should extend the components in a controlled way, so that it does not include undesirable features.

All the mechanisms assume that the component logics are presented in the same way. In technical terms we say they are homogeneous. For instance, both of them are presented by Hilbert calculi. However, some assume that component logics need some preparation before being combined. For instance, assume that we say that component logics are endowed with an algebraic semantics. In this case, we have to say how the semantic structures of the component logics induce an algebraic semantics. In the book we deal with heterogeneous fibring in a moderated way in Chapter 3 and with heterogeneous fibring of deductive systems in Chapter 4.

One of the most challenging problems is related to proving transference results. That is, to investigate sufficient conditions for the preservation of properties, namely soundness, completeness, decidability, consistency, interpolation, from the components into the resulting logic.

Combination mechanisms can be extended to a finite number of components and sometimes even to an infinite number of components.

Among the different combination mechanisms we can refer to fibring which is one of the objects of this book. Fusion, if not historically the first, is the simplest method, and the best studied combination mechanism.

Combining logics in the perspective of this book does not mean only synthesizing or composing logics (which is called splicing), but is also intended to work in the opposite direction of decomposing logics, called splitting. Herein, we analyze the possible-translations semantics mechanism.

The idea of writing this book originated during *The Workshop on Combination of Logics: Theory and Applications* (CombLog04) [50, 52], held in the Center for

Logic and Computation, at the Department of Mathematics of IST, Technical University of Lisbon, Portugal, from July 28-30, 2004. Encouraged by the vigor of the field and by the interest triggered by this and several other conferences (such as [234, 112, 162, 9, 138, 10, 11]), we decided to accept the challenge to produce a book containing some basic ideas, methods and techniques that could help logicians, computer scientists and philosophers to have access to a general yet elementary theory of combinations of logics. The book intended to bring together a sample of results, problems and perspectives involving the idea of cutting and pasting logics, explaining when possible the role of the underlying constructions as universal arguments in the categorial sense.

We depart here from a basic universe of logic systems starting with propositional-based systems endowed with Hilbert calculi and ordered algebraic semantics. This basic setting is already rich enough to encompass interesting features of fibring with several applications and to provide the basic techniques for the trade of combining systems. Later on we extend the notion to the first-order and to the higher-order domains.

Chapter 1 is an introductory overview to the essential ingredients of composing and decomposing logics. In Section 1.1, we introduce the concept of consequence system as the basic abstraction to describe a logic system. In Section 1.2, we present the basic ideas about composing or splicing logics and decomposing or splitting logics. We also introduce a technical summary of some combination mechanisms like fusion, product and fibring by functions of modal logics. We also refer to Gödel provability logic as an illustration of a splitting mechanism. In Section 1.3, we provide a very brief introduction to algebraic fibring using Hilbert calculi. In Section 1.4, we sketch the splitting mechanism called possible-translations semantics.

Chapter 2 concentrates on fibring of propositional based logics presented as Hilbert calculi. Moreover, some preservation results are introduced. In Section 2.1, signatures and their fibring are presented. In Section 2.2, we dedicate our attention to the fibring of Hilbert calculi. We illustrate the concepts with several examples including classical logic, modal logics, intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 2.3, we discuss several preservation results. Finally, Section 2.4 presents some final remarks.

Chapter 3 is dedicated to the fibring of semantics for propositional based logics. Ordered algebras are the basic semantic structures adopted. We also include the relationship to fusion and fibring by functions. Again some preservation results are given. In Section 3.1, we introduce the semantic structures and their fibring. We illustrate the concepts with several examples including classical logic, modal logics, intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 3.2, we present the notions of logic system, soundness and completeness. In Section 3.3, we discuss the preservation of soundness and completeness properties. In Section 3.4, we establish the relationship between the present approach and fibring by functions. In Section 3.5 we present some final comments.

Chapter 4 is dedicated to the analysis of fibring of logics that are not presented in the same way. Two solutions are proposed. The first one is based on fibring of consequence systems and the second one on abstract proof systems. Some preservation results are established. Section 4.1 concentrates on fibring of consequence systems using a fixed point operator. Several examples are given for logics presented either in a proof-theoretic or a model-theoretic way. Section 4.2 focuses on the notion of abstract proof system and looks at the proof systems induced by Hilbert, sequent and tableau calculi. Moreover, it includes the notion of fibring of abstract proof systems. We also discuss some relationships between consequence systems and proof systems. In Section 4.3 we present some final remarks.

Chapter 5 studies composition of non-truth functional logics via fibring, an important extension of the theory, considering that many of the interesting logics for applications are not truth-functional. In Section 5.1, the notion of interpretation system presentation is introduced. In Section 5.2 the notions of unconstrained and constrained fibring of interpretation system presentations is defined. In Section 5.3 we again use Hilbert calculus as the suitable proof-theoretic notion. In Section 5.4 some preservation results are established, namely, the preservation of soundness and completeness. Section 5.5 discusses self-fibring in the context of non-truth-functional logics. In Section 5.6 we present some final comments.

Chapter 6 concentrates on fibring of first-order based logics. It can be seen as an extension of the fibring of propositional based logics, choosing particular powerset algebras as semantic structures. The running example is fibring of classical first-order logic and modal logic. In Section 6.1, first-order based signatures and the corresponding languages are introduced. Next, in Section 6.2, we present interpretation structures and interpretation systems. First-order Hilbert calculi are presented in Section 6.3. Section 6.4 introduces first-order logic systems. Then, in Section 6.5, we define fibring of first-order based logics. The preservation of completeness and other metatheorems by fibring is discussed in Section 6.6, where we also briefly sketch a proof of completeness for a particular class of first-order logic systems. In Section 6.7 we make some final remarks.

Chapter 7 deals with higher-order quantification logics. The semantic structures are generalizations of the usual topos semantics for higher-order logics. In Section 7.1 we introduce the relevant signatures. In Section 7.2 the Hilbert calculus is presented. In Section 7.3 is dedicated to setting up the semantic notions. Section 7.4 introduces the notion of logic system, and we briefly discuss some related notions such as soundness and completeness. The novelty here is that the usual notion of soundness must be modified in the present framework. In Section 7.5, a general completeness theorem is established. In Section 7.6, the notions of constrained and unconstrained fibring of logic systems are given, and it is shown that soundness is preserved by fibring and a completeness preservation result is obtained. In Section 7.7 we briefly discuss the main results described in the chapter.

In Chapter 8, we turn our attention to modulated fibring. This variant was developed to cope with collapsing problems: in some cases when two logics are combined one of them collapses with the other. We illustrate the concepts with examples including propositional logic, intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 8.1, we introduce the notions of modulated signature and modulated signature morphisms. In Section 8.2, we describe modulated interpretation structures, modulated interpretation systems and the corresponding morphisms. Next we present the notion of bridge between modulated interpretation systems. In Section 8.3, we define modulated Hilbert calculus and their morphisms. In Section 8.4 is dedicated to modulated logic systems and their corresponding morphisms. In Section 8.5, we establish soundness and completeness preservation results. Finally, Section 8.6 presents some final comments.

Chapter 9 introduces the problem of splitting logics, emphasizing the role of possible-translations semantics and contrasting with the previous chapters that deal with forms of splicing. In Section 9.1, a category of propositional based signatures suitable for splitting logics is introduced, as well as the corresponding category of consequence systems. In Section 9.2 the technique known as possible-translations characterization is analyzed, and some applications are given. In Section 9.3 two methods for combining matrix logics, plain fibring and direct union of matrices, are reviewed. Finally, Section 9.4 presents some final comments.

In Chapter 10 we discuss new tendencies on fibring. In Section 10.1, we motivate network fibring using modal logic. In Sections 10.2, 10.3, 10.4 and 10.5, some case-studies are introduced. Section 10.2 discusses integration of information flows by describing a system in which reasoning and proofs from different sources of information can be accommodated. In Section 10.3, we refer to some generalizations of logic input/output operations. We also discuss how to combine input/output operations into networks. In Section 10.4, we discuss the fibring of neural networks. In Section 10.5, we turn our attention to recursive Bayesian networks. In Section 10.6, the notion of self-fibring of networks is introduced. Section 10.7 presents some concluding remarks.

Finally, in Chapter 11 we first present a summing-up of the different techniques for combining logics presented in this book together with their main features. Then we move to a brief overview of applications where fibring can be directly used, as well as to emergent fields of application. It also includes an outlook of new research directions in both the existing combination mechanisms but also to new forms of combination.

We observe that most chapters of the book deal with combination of logics rather than with decomposition of logics. This happens because splitting mechanisms are not so well developed.

We assume that the readers are familiar with basic logic notions of classical propositional and first-order logics at the level of, for instance, [206] and [90] and propositional modal logics at the level of, for instance, [153]. Although not mandatory, a very basic knowledge of categories (see [186]) is useful for better understanding the minimality of the constructions.

The book is intended to be a research monograph for those that want to know the state-of-the art in composing and decomposing logics, that want to know about issues worthwhile to be pursued, as well as potential contemporary applications of these techniques. If you are one of these we recommend that you have the patience to read the whole book. If you want to focus on particular aspects of combination of logics, we suggest several paths hoping that one of them is of your taste.

- If the reader is only interested in knowing what are the main issues in the combination of logics, we recommend Chapters 1, 2 and 3 which provide a basic account on consequence systems and the basic notions of propositional fibring;
- If you are curious about decomposition and its importance to non-truth functional logics you should read Chapters 1 and 9 and maybe it is useful reading Chapter 5;
- The reader interested in a more general form of fibring (capturing more propositional-based logics) and avoiding the well known collapsing problem should concentrate on Chapter 8, besides Chapters 1, 2 and 3;
- Someone with research interest in proof systems and how to combine different proof systems should read Chapters 1, 2 and 4;
- If your interests are in modal logic, we suggest you read Chapters 1, 2, 3, 4 and 6;
- If your interests are in hybrid logic and labeled deduction in general, we suggest you read Chapters 1, 2, 3, 4 and 10;
- If you are a first-order logician and have curiosity on combination of logics, we suggest you read Chapters 1, 2, 3 and, more importantly, Chapter 6;
- If you are a higher-order logician and want to grasp what is combination of logics, we suggest you read Chapters 1, 2, 3 and, of course, Chapter 7;
- If you are an intuitionistic logician and would like to know about combination of logics, we suggest you read Chapters 1, 2, 3 and Chapter 8;
- If you like to know the potential of combination in contexts that are not logical in nature you should read Chapter 10.

For a summing-up of the techniques used in the book, as well as some applications and topics for further research, we recommend Chapter 11.

Acknowledgments

The authors are in debt to their friends and colleagues that participated in FibLog project: Amílcar Sernadas, Carlos Caleiro, Jaime Ramos, João Rasga, Luís Cruz-Filipe and Paulo Mateus.

A special thanks is due to Amílcar Sernadas whose ideas and work is present in many of the chapters of the book. Also to Carlos Caleiro who collaborated in the comparison between fibring by functions and algebraic fibring and in the fibring of non-truth functional logics, to Alberto Zanardo for his collaboration in the first proof of completeness preservation by fibring and fibring of first-order based logics, to João Rasga for his collaboration in the definition of modulated fibring and preservation of interpolation, to Luís Cruz-Filipe for his collaboration in heterogeneous fibring and to Víctor Fernández for his collaboration in the definition of plain fibring. We are thankful to Paulo Mateus for his help with security issues, in particular, related to zero-knowledge protocols. We are grateful to Amílcar Sernadas and Paulo Mateus for their advise on quantum related issues. We also acknowledge work with Luca Viganò although not directly reflected in this book. We are in debt to Miguel Dionísio for his precious help with several LaTeXproblems.

We would like to thank the Security and Quantum Information Security Group at Instituto de Telecomunicações (SQIG - IT) and Center for Logic and Computation (CLC) and all the colleagues there. This research was supported by Fundação para a Ciência e a Tecnologia (FCT) and FEDER, namely through CLC, IT and the FibLog POCTI/2001/MAT/37239, the QuantLog POCI/MAT/55796/2004 and the KLog PTDC/MAT/68723/2006 Projects. This research was also supported by The State of São Paulo Research Foundation (FAPESP), Brazil, through the Thematic Project number 2004/14107-2 ("ConsRel"). Walter Carnielli and Marcelo Coniglio also acknowledge support from The National Council for Scientific and Technological Development (CNPq), Brazil, through individual research grants.

Lisbon, July of 2007
The Authors

Chapter 1

Introductory overview

It is not an easy matter to trace down the origins of the idea of combining reasoning (in a schematic or semi-formal manner) and relations upon them (by means of diagram, rules or other mechanisms). In a sense this has origins in the history of European philosophy itself: Plato's dialogue *Sophist* inquires about the methods of philosophy, and part of his conclusions involves the limitations of common language, and the danger of using common language which may lead to fallacious conclusions. For example, Plato shows that "not being" is a form of "being", a confusion that common language cannot help to cope with.

On the other hand, logic is the branch of knowledge where language receives the highest systematized treatment. Concepts, such as time, belief, knowledge, inheritance, relevance and dependence, their mutual relations and the methods to draw conclusions from them can be expressed in the most convenient way. Contemporary logic has pushed this tendency to extremes, with not entirely positive consequences, in the opinion of critiques (see [267]). So, in a sense, if the logical analysis taken to an extreme has separated concepts and methods, we wonder why not join them together.

Under the light of today's logic, this can be achieved with greater accuracy. This is precisely the object of this book: how to cut and paste logics and how to use them. But the roots of the idea are much older. The Catalan mystic and logician Ramón Llull, born in 1235, used logic and mechanical methods based on symbolic notation and combinatorial diagrams to relate, he thought, all forms of knowledge. His main work, the *Ars Magna*, makes him, at the same time, a precursor of combinatorics and of the art of combining logics. The method of Llull consisted of a series of concentric circles with attributes such as *bonitas* (goodness), *magnitudo* (greatness), *eternitas* (eternity), and categories such as *flame*, *man*, *animal*, which could be combined with, for example *where?*, *why?* and *how?*. By combining them, one could (at least try) to solve riddles such as "Where does the flame go when a candle is put out?".

Llull inaugurated the idea of a "philosophical machine" and had a great influence on Gottfried Leibniz. According to Couturat in [64], Leibniz was the first to see explicitly the possibilities of applying Llull's methods to formal logic in his *Dissertatio de Arte Combinatoria*, where exhaustive techniques to combine premises and conclusions in the language of Aristotelian syllogisms was considered. Later on, the British economist and logician William Steven Jevons, famous for the invention of a "logic machine" to draw syllogistic conclusions, used similar ideas in his "logical alphabet". From some point on, the attempts to combine logical devices lead to the hardware side of constructing machines. The reader is invited to see the excellent book by Martin Gardner [121] for the account of the "demonstrator" of Charles Stanhope and of the logic machines of Allan Marquand.

If, in the question above, the terms are appropriately changed to *knowledge*, *belief*, *time*, *tense*, and the question by "Where does the learning go when knowledge is separated from time" we can foresee how combining logics would impact philosophical logic, knowledge representation, artificial intelligence, cognitive sciences and so many other areas of interest to philosophers, linguists and computer scientists.

The use of formal logic for representing conceptual reasoning favored the emergence of the so-called "non-classical" logics in the first quarter of the XX century, as opposed to "classical" logic, usually understood as two-valued propositional logic and predicate logic with equality. The label "non-classical" is far from appropriate, since, for example, the logical properties of necessity, possibility, impossibility, contingency and related concepts were extensively treated by Aristotle (see [135]) and other ancient authors. This tradition, including nowadays the logical properties of permission, prohibition, belief, knowledge, tense, and many other evolved to what is called *modal logic*. Besides modal, there are today dozens of such "non-classical" logics such as many-valued logics, paraconsistent logics, intuitionistic logics, dependence logics and relevant logics.

If logic is objective, how can there be so many logics? This intriguing question is posed in [88], and the proposed answer, in the same book, is that what one pays attention to, in reasoning, is what determines which logic is appropriate. So logic, as a discipline, is subject-matter dependent. Classical logic is appropriate to reasoning with purely mathematical concepts, such as points, lines, sets, numbers, groups, equations and topological spaces, where no direct influence of time, intention, intensity, purpose, cause or effect is taken into account. On the other hand, the so-called non-classical logics, emerge when specialized domains are taken into consideration, and one pays attention to specific constituents.

Thus, many-valued logics may be pertinent when one needs to pay attention to degrees of truth. Paraconsistent logics may be appropriate when one needs to reason under contradictions but avoiding trivial theories. Dependence logics may be suited if one prefers to see propositions as possessing referential content. Relevant or relevance logics may be apropos if one is interested in how assumptions are related to conclusions in derivations. Modal logics may be the right choice if one is engaged on reasoning with necessity, possibility, knowledge, belief, permission, prohibition and obligation. Intuitionistic logic may be the case if one is occupied with some aspects of constructiveness in argumentation. And so on and so forth.

Now it seems immediately obvious that reality is many-faced. A concrete question may involve several aspects that one wishes to pay attention to, and a combination of pre-existing logics to reason with such a question would be the best decision. Instead of building a new logic from scratch, it may be wiser to depart from the assumption that logics can be reusable and assembled in new and convenient manners.

However, one must face the need to integrate distinct languages and inference engines. Different families of symbols have to be merged, and sometimes the same symbol in different logics has a different meaning. Moreover, derivations can have a completely different nature in different logics.

This chapter provides an overview, aiming to anticipate, in a very simplified form, some aspects that will be pursued in full detail in the next chapters. As an appetizer served before the function, it will show the issues in lesser content.

The contents of this chapter are as follows. In Section 1.1, we present consequence systems as an abstract way to present logics via a consequence operator. We also introduce the concept of morphism to relate consequence systems. In Section 1.2, we present the basic ideas about composing or splicing logics and decomposing or splitting logics. We also introduce a technical summary of the most relevant mechanisms for splicing and splitting, namely fusion, product and fibring by functions of modal logics and Gödel provability logic. In Section 1.3, we provide an introduction to algebraic fibring from a deductive perspective. We also motivate that our aim is to define composing and decomposing mechanisms as minimal or maximal constructions. The issues of this section are discussed in detail in most of the chapters of this book. In Section 1.4, we give an introduction to the splitting mechanism called possible-translations semantics. A deeper account of this technique is explored in Chapter 9.

1.1 Consequence systems

A fundamental question, previous to any attempt to combining logics, is how to define the logics which are to be combined. There are many authors that tried to answer this question. The interested reader can take a look at [103]. Herein, a logic is a consequence system following the formulation given by Alfred Tarski (see [258]). The quest for the abstract definition of logic, as a consequence operator, seems to go back to Bernard Bolzano (see [266] and also [231] in [118]).

A consequence system is usually a pair composed by a set, the language or the set of formulas, and a map indicating for each subset of formulas the respective consequences. Some requirements are imposed on the map depending on the objectives and the properties of the logic at hand. Consequence systems can be defined in a proof-theoretic way or in a model-theoretic way.

We start with some notation. Let S be a set. We denote by $\wp S$ the set of all subsets of S and by $\wp_{\text{fin}} S$ the set of all finite subsets of S.

Definition 1.1.1 A *consequence system* is a pair

$$\langle L, \mathsf{C} \rangle$$

where L is a set and $\mathsf{C} : \wp L \to \wp L$ is a map satisfying:

- $\Gamma \subseteq \mathsf{C}(\Gamma)$ — extensiveness
- if $\Gamma_1 \subseteq \Gamma_2$ then $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_2)$ — monotonicity
- $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\Gamma)$ — idempotence

∇

The set L is the language, that is, the set of formulas. The map $\mathsf{C} : \wp L \to \wp L$ is a closure operator [198] called consequence operator. For each $\Gamma \subseteq L$, $\mathsf{C}(\Gamma)$ is the set of *consequences* of the set of *hypotheses or assumptions* Γ. Extensiveness means that an hypothesis in a set is a consequence of this set. Monotonicity states that a formula that is a consequence of a set of hypotheses is also a consequence of any bigger set of hypotheses. Idempotence means that we can use lemmas to obtain consequences of a set of formulas.

An alternative characterization of the consequence operator can be given.

Proposition 1.1.2 *A map $\mathsf{C} : \wp L \to \wp L$ is a consequence operator if and only if it satisfies the following properties:*

- $\Gamma \subseteq \mathsf{C}(\Gamma)$ — extensiveness
- $(\mathsf{C}(\Gamma_1) \cup \mathsf{C}(\Gamma_2)) \subseteq \mathsf{C}(\Gamma_1 \cup \Gamma_2)$ — preservation of unions
- *If* $\Gamma_2 \subseteq \mathsf{C}(\Gamma_1)$ *and* $\Gamma_3 \subseteq \mathsf{C}(\Gamma_2)$ *then* $\Gamma_3 \subseteq \mathsf{C}(\Gamma_1)$ — transitivity.

Proof. Let C be a consequence operator. We start by proving preservation by unions. We have that $\Gamma_1 \subseteq \Gamma_1 \cup \Gamma_2$ and $\Gamma_2 \subseteq \Gamma_1 \cup \Gamma_2$. Then $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_1 \cup \Gamma_2)$ and $\mathsf{C}(\Gamma_2) \subseteq \mathsf{C}(\Gamma_1 \cup \Gamma_2)$ by monotonicity and so $\mathsf{C}(\Gamma_1) \cup \mathsf{C}(\Gamma_2) \subseteq \mathsf{C}(\Gamma_1 \cup \Gamma_2)$. Now we prove transitivity. Assume that $\Gamma_2 \subseteq \mathsf{C}(\Gamma_1)$ and $\Gamma_3 \subseteq \mathsf{C}(\Gamma_2)$. Then, $\mathsf{C}(\Gamma_2) \subseteq \mathsf{C}(\mathsf{C}(\Gamma_1))$ by monotonicity. Hence, $\Gamma_3 \subseteq \mathsf{C}(\mathsf{C}(\Gamma_1))$ and so $\Gamma_3 \subseteq \mathsf{C}(\Gamma_1)$ by idempotence.

Assume now that C satisfies extensiveness, preservation of unions and transitivity. We start by proving monotonicity. Assume that $\Gamma_1 \subseteq \Gamma_2$. Then, $\Gamma_1 \cup \Gamma_2 = \Gamma_2$. Hence, $\mathsf{C}(\Gamma_1 \cup \Gamma_2) = \mathsf{C}(\Gamma_2)$. Therefore, by preservation of unions, $\mathsf{C}(\Gamma_1) \cup \mathsf{C}(\Gamma_2) \subseteq \mathsf{C}(\Gamma_2)$ and so $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_2)$. Finally, we prove idempotence. Since $\mathsf{C}(\Gamma) \subseteq \mathsf{C}(\Gamma)$ and $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\mathsf{C}(\Gamma))$ then $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\Gamma)$ by transitivity. $\triangleleft$

This presentation of the consequence operator, namely transitivity, is closer to the initial notion given by Bolzano (see [266]).

It is worthwhile to observe that we do not have in general that $\mathsf{C}(\emptyset) = \emptyset$ and that $\mathsf{C}(\Gamma_1 \cup \Gamma_2) \subseteq \mathsf{C}(\Gamma_1) \cup \mathsf{C}(\Gamma_2)$. Hence, the consequence operator is not in general a Kuratowski operator (see [160]). For more details about the relationship between logic and closure spaces see [198]. See also [280].

A consequence system is said to be *compact* or *finitary* if

$$\mathsf{C}(\Gamma) = \bigcup_{\Phi \in \wp_{\mathrm{fin}} \Gamma} \mathsf{C}(\Phi).$$

Compact consequence systems are also known as *algebraic* systems (as in [77]).

Example 1.1.3 An example of a consequence system is $\langle L_{\mathbb{P}}, \mathsf{C}\rangle$ where $L_{\mathbb{P}}$ is the set of propositional formulas over the set $\mathbb{P}$ and $\mathsf{C}(\Gamma)$ is the set of all formulas that are derived from Γ using a Hilbert calculus for classical propositional logic.

Another example of a consequence system is $\langle L'_{\mathbb{P}}, \mathsf{C}'\rangle$ where $L'_{\mathbb{P}}$ is the set of modal propositional formulas over the set $\mathbb{P}$ and $\mathsf{C}'(\Gamma)$ is the set of all formulas that are derived from Γ using a Hilbert calculus for modal propositional logic. ▽

Another characterization of consequence operators can be given. But first we prove the following auxiliary result.

Lemma 1.1.4 *Let $\langle L, \mathsf{C}\rangle$ be a consequence system. Then*

$$\mathsf{C}(\Gamma) = \mathsf{C}(\mathsf{C}(\Gamma))$$

for every $\Gamma \subseteq L$.

Proof. Use the idempotence for one side and extensiveness and monotonicity conditions for the other. ◁

We note that this property is in fact idempotence in the usual algebraic sense.

Proposition 1.1.5 *The map $\mathsf{C} : \wp L \to \wp L$ is a consequence operator if and only if the following condition holds: (a) $\Phi \subseteq \mathsf{C}(\Gamma)$ if and only if (b) $\mathsf{C}(\Phi) \subseteq \mathsf{C}(\Gamma)$, for every $\Gamma, \Phi \subseteq L$.*

Proof. Assume that $\mathsf{C} : \wp L \to \wp L$ is a consequence operator. Assume also that $\Phi \subseteq \mathsf{C}(\Gamma)$. By monotonicity $\mathsf{C}(\Phi) \subseteq \mathsf{C}(\mathsf{C}(\Gamma))$ and so $\mathsf{C}(\Phi) \subseteq \mathsf{C}(\Gamma)$ follows by idempotence. Suppose that $\mathsf{C}(\Phi) \subseteq \mathsf{C}(\Gamma)$, then using extensiveness $\Phi \subseteq \mathsf{C}(\Phi)$, we get $\Phi \subseteq \mathsf{C}(\Gamma)$.

Conversely, assume that the condition holds. We prove that C is a consequence operator. Extensiveness: from $\mathsf{C}(\Gamma) \subseteq \mathsf{C}(\Gamma)$ we get $\Gamma \subseteq \mathsf{C}(\Gamma)$ using the fact that (b) implies (a). Monotonicity: using extensiveness $\Gamma_1 \subseteq \Gamma_2 \subseteq \mathsf{C}(\Gamma_2)$ and so $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_2)$ using the fact that (a) implies (b). Idempotence: from $\mathsf{C}(\Gamma) \subseteq \mathsf{C}(\Gamma)$ we get $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\Gamma)$, using the fact that (a) implies (b). ◁

We observe that consequence systems do not cover every possible logic that one can think. Namely, the concept leaves outside logics where sets of formulas are not considered like for instance in linear logic [125]. It seems that for covering this kind of logics one needs a more general notion.

Consequence systems can be related. We introduce a weakness relation between consequence systems.

Definition 1.1.6 The consequence system $\langle L, \mathsf{C} \rangle$ is *weaker* than consequence system $\langle L', \mathsf{C}' \rangle$, written

$$\langle L, \mathsf{C}\rangle \leq \langle L', \mathsf{C}'\rangle$$

if $L \subseteq L'$ and $\mathsf{C}(\Gamma) \subseteq \mathsf{C}'(\Gamma)$ for every subset Γ of L. $\triangledown$

We also say that a consequence system $\langle L, \mathsf{C} \rangle$ is *partially weaker* than consequence system $\langle L', \mathsf{C}' \rangle$, written

$$\langle L, \mathsf{C}\rangle \leq_p \langle L', \mathsf{C}'\rangle$$

if $L \subseteq L'$ and $\mathsf{C}(\emptyset) \subseteq \mathsf{C}'(\emptyset)$. When C is a syntactic operator, this means that all theorems of C are also theorems of C' and when C is a semantic operator, this means that all valid formulas of C are also valid formulas of C'. Of course, if $\mathsf{C} \leq \mathsf{C}'$ then $\mathsf{C} \leq_p \mathsf{C}'$ but not the other way around.

Example 1.1.7 Recall the consequence systems $\langle L_{\mathbb{P}}, \mathsf{C}\rangle$ and $\langle L'_{\mathbb{P}}, \mathsf{C}'\rangle$ presented in Example 1.1.3. We have that $\langle L_{\mathbb{P}}, \mathsf{C}\rangle \leq \langle L'_{\mathbb{P}}, \mathsf{C}'\rangle$. $\triangledown$

We can also relate consequence systems that have completely different languages using morphisms.

Definition 1.1.8 A *consequence system morphism* $h : \langle L, \mathsf{C} \rangle \to \langle L', \mathsf{C}' \rangle$ is a map $h : L \to L'$ such that

$$h(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}'(h(\Gamma))$$

for every $\Gamma \subseteq L$. $\triangledown$

That is, the image of every consequence of a set of formulas is a consequence of the image of the set. We observe that the converse is not always true, namely when $h : L \to L'$ is not injective. Consequence systems and their morphisms constitute a category **Csy**.

Note that when $\langle L, \mathsf{C}\rangle \leq \langle L', \mathsf{C}'\rangle$ then there is a consequence system morphism from $\langle L, \mathsf{C}\rangle$ to $\langle L', \mathsf{C}'\rangle$ where the map from L to L' is just an inclusion.

An alternative characterization is as follows. We start by introducing some notation. Given a map $h : L \to L'$, we denote by $h^{-1}(\Gamma')$ the set of formulas $\{\gamma \in L : h(\gamma) \in \Gamma'\}$ for each $\Gamma' \subseteq L'$.

Proposition 1.1.9 *Let $\langle L, \mathsf{C}\rangle$ and $\langle L', \mathsf{C}'\rangle$ be consequence systems. A map between formulas $h : L \to L'$ is a consequence system morphism $h : \langle L, \mathsf{C}\rangle \to \langle L', \mathsf{C}'\rangle$ if and only if*

$$\mathsf{C}(h^{-1}(\Gamma')) \subseteq h^{-1}(\mathsf{C}'(\Gamma'))$$

for every $\Gamma' \subseteq L'$.

Proof. Assume that h is a consequence system morphism. Let $\varphi \in \mathsf{C}(h^{-1}(\Gamma'))$. Then $h(\varphi) \in h(\mathsf{C}(h^{-1}(\Gamma')))$ and since h is a morphism $h(\varphi) \in \mathsf{C}'(h(h^{-1}(\Gamma')))$. On the other hand, $h(h^{-1}(\Gamma')) \subseteq \Gamma'$ hence, by monotonicity, $\mathsf{C}'(h(h^{-1}(\Gamma'))) \subseteq \mathsf{C}'(\Gamma')$. Therefore, $h(\varphi) \in \mathsf{C}'(\Gamma')$ and so $\varphi \in h^{-1}(\mathsf{C}'(\Gamma'))$.
Assume now that $\mathsf{C}(h^{-1}(\Gamma')) \subseteq h^{-1}(\mathsf{C}'(\Gamma'))$ for every Γ'. Let $\varphi \in \mathsf{C}(\Gamma)$. Then $\varphi \in \mathsf{C}(h^{-1}(h(\Gamma)))$. By the hypothesis, $\mathsf{C}(h^{-1}(h(\Gamma))) \subseteq h^{-1}(\mathsf{C}'(h(\Gamma)))$, hence $\varphi \in h^{-1}(\mathsf{C}'(h(\Gamma)))$ (since $\Gamma \subseteq h^{-1}(h(\Gamma))$ and so $\mathsf{C}(\Gamma) \subseteq \mathsf{C}(h^{-1}(h(\Gamma)))$ by monotonicity), then $h(\varphi) \in h(h^{-1}(\mathsf{C}'(h(\Gamma))))$ and so $h(\varphi) \in \mathsf{C}'(h(\Gamma))$. ◁

This characterization is of course related to the notion of continuous map in topological spaces (see [160]).

It is worthwhile to say what is the union of consequence systems. We will see below that most combination mechanisms (including for instance fibring) do not correspond to the union of consequence systems.

Definition 1.1.10 Let $\{\mathsf{C}_i\}_{i\in I}$, where $\mathcal{C}_i = \langle L_i, \mathsf{C}_i\rangle$, be a family of consequence systems. Their *union* is the following consequence system:

$$\mathcal{C} = \langle \bigcup_{i\in I} L_i, \mathsf{C}\rangle$$

where $\mathsf{C}(\bigcup_{i\in I}\Gamma_i)$ is $\bigcup_{i\in I}\mathsf{C}_i(\Gamma_i)$. ▽

Instead of presenting consequence via an operator we can look at consequence as a binary relation between the set of all sets of formulas and the set of formulas. Given a binary relation $S \subseteq A \times B$, we may write aSb whenever $\langle a, b\rangle \in S$.

Then, a *consequence relation* over L is a set

$$\mathsf{R} \subseteq \wp L \times L$$

satisfying the following postulates: (i) if $\varphi \in \Gamma$ then $\Gamma\mathsf{R}\varphi$; (ii) if $\Psi\mathsf{R}\varphi$ e $\Gamma\mathsf{R}\psi$ for every $\psi \in \Psi$ then $\Gamma\mathsf{R}\varphi$; (iii) if $\Gamma_1\mathsf{R}\varphi$ and $\Gamma_1 \subseteq \Gamma_2$ then $\Gamma_2\mathsf{R}\varphi$. A consequence operator $\mathsf{C} : \wp L \to \wp L$ induces a consequence relation R_C such that, for every $\Gamma \subseteq L$ and every $\varphi \in L$,

$$\Gamma\mathsf{R}_\mathsf{C}\varphi \text{ if and only if } \varphi \in \mathsf{C}(\Gamma).$$

A consequence relation R induces a consequence operator C_R such that, for every $\Gamma \subseteq L$,

$$\mathsf{C}_\mathsf{R}(\Gamma) = \{\varphi \in L : \Gamma\mathsf{R}\varphi\}.$$

It is worth noting that the map $\mathsf{C} \to \mathsf{R}_\mathsf{C}$ is the inverse of the map $\mathsf{R} \to \mathsf{C}_\mathsf{R}$, and vice-versa. Thus, a consequence system can be defined indistinctly as a pair $\langle L, \mathsf{C} \rangle$ such that C is a consequence operator over L, or as a pair $\langle L, \mathsf{R} \rangle$ such that R is a consequence relation over L. Note that $\langle L, \mathsf{R} \rangle$ is weaker than $\langle L', \mathsf{R}' \rangle$ if and only if $L \subseteq L'$ and $\mathsf{R} \subseteq \mathsf{R}'$.

Sometimes we may write Γ^R instead of $\mathsf{C}_\mathsf{R}(\Gamma)$.

For the purpose of most combination mechanisms, the consequence systems are such that their language is always generated from a family of connectives. Hence, we do not include L in the definition of a consequence system, but instead we use a signature C defining the family of connectives in each case. As an illustration we define propositional based signatures and the induced languages.

Definition 1.1.11 A *propositional based signature* is any family of sets

$$C = \{C_k\}_{k \in \mathbb{N}}$$

such that $C_i \cap C_j = \emptyset$ if $i \neq j$. ∇

The elements of the set C_k are called *k-ary connectives* or connectives of *arity* k. In particular, the elements of C_0 are called *constants*.

We will consider unions of propositional signatures. Given propositional signatures C' and C'', their union is the signature

$$C' \cup C''$$

where $(C' \cup C'')_k = C'_k \cup C''_k$ for each $k \in \mathbb{N}$. We use $C' \setminus C''$ to denote the signature such that $(C' \setminus C'')_k = C'_k \setminus C''_k$ for each $k \in \mathbb{N}$.

Given two signatures C e C', we say that C *is contained in* C', denoted by

$$C \leq C'$$

if $C_k \subseteq C'_k$, for every $k \in \mathbb{N}$. In some situations, we would like to include a set $\mathbb{P}$ of propositional constants. Then we would say that $\mathbb{P} \subseteq C_0$.

We assume that Ξ is a set of schema variables. Schema variables play an important role when combining logics, in particular for deduction, as we explain in Chapter 2.

Definition 1.1.12 Let $C = \{C_k\}_{k \in \mathbb{N}}$ be a signature. The propositional based *language generated by* C is the set $L(C)$ defined as the least set $L(C)$ that satisfies the following properties:

- $\Xi \subseteq L(C)$;
- $(c(\varphi_1, \ldots, \varphi_k)) \in L(C)$ whenever $k \in \mathbb{N}$, $c \in C_k$ and $\varphi_1, \ldots, \varphi_k \in L(C)$. ∇

The elements of $L(C)$ are called *formulas* over C. In particular, $C_0 \subseteq L(C)$.

Typical deductive systems such as Hilbert, tableau, sequent and natural deduction calculi induce consequence systems. We can look at each kind of deductive system as a presentation of a consequence system. Observe that it is quite common to work with these presentations instead of working with the consequence systems themselves. That is the case of most chapters in this book.

We illustrate how a Hilbert calculus induce a consequence system. We need to define Hilbert calculus, substitution and derivation.

Definition 1.1.13 A *Hilbert calculus* is a pair

$$H = \langle C, R \rangle$$

such that:

- C is a signature;
- R is a set inference rules, that is, a set of pairs $\langle \Delta, \psi \rangle$ where $\Delta \subseteq L(C)$ is a finite set and $\psi \in L(C)$. ∇

When $\Delta = \emptyset$ we say that the inference rule is an axiom. Otherwise it is said to be a rule. Sometimes when introducing axioms we may, for simplicity, indicate only the formula. We now define the notion of derivation in a Hilbert calculus. Before we have to introduce the notion of substitution.

The objective of a substitution is to replace schema variables by formulas. A *substitution* is a map

$$\sigma : \Xi \to L(C).$$

Substitutions can be extended to formulas in a natural way. We denote by $\sigma(\varphi)$ the formula that results from substituting each schema variable ξ in φ by $\sigma(\xi)$. Moreover, substitutions can be extended to sets of formulas. We denote by $\sigma(\Gamma)$ the set of formulas $\{\sigma(\gamma) : \gamma \in \Gamma\}$.

Definition 1.1.14 A *derivation* in H from a set $\Gamma \subseteq L(C)$ is a sequence

$$\varphi_1 \cdots \varphi_n$$

such that for $i = 1, \ldots, n$ each φ_i is either an element of Γ or there is a substitution σ and an inference rule $\langle \Delta, \psi \rangle$ in H such that $\sigma(\psi)$ is φ_i and $\sigma(\delta)$ is φ_j for some $j < i$, for every $\delta \in \Delta$.

We also say that φ_n is *derived from* Γ and use the following notation

$$\Gamma \vdash_H \varphi_n.$$

∇

A Hilbert calculus H induces a consequence system

$$\mathcal{C}(H) = \langle C, \vdash_H \rangle$$

where, for each $\Gamma \subseteq L(C)$, $\Gamma^{\vdash_H}$ is the set $\{\varphi \in L(C) : \Gamma \vdash_H \varphi\}$. Observe that this consequence system is compact. It is also structural in the following sense.

A consequence system is said to be *structural* if, for every substitution σ we have that:

$$\sigma(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\sigma(\Gamma)).$$

That is, if a consequence system $\mathcal{C}$ is structural, then σ is a consequence system endomorphism (from $\mathcal{C}$ to $\mathcal{C}$).

Consequence systems that are compact and structural are called *standard* in the terminology of [280]. The notions of compact, structural and standard consequence systems can be expressed in terms of consequence relations in the obvious way.

Similarly, semantic entailments are also consequence operators. For instance, propositional entailment associated with valuations and modal entailment associated to Kripke structures are examples of consequence operators. Semantic structures can also be seen as presentations of the semantic entailment.

Consequence operators are relevant in all chapters of the book. In Chapter 2, we introduce a Hilbert consequence operator generated by a Hilbert calculus. In Chapter 3, we introduce a semantic consequence operator. We also introduce the notion of soundness, as saying that the set of Hilbert consequences is included in the set of semantic consequences, and the notion of completeness, as stating that the set of semantic consequences is included in the set of Hilbert consequences. In Chapter 5, consequence systems are used in another way. When considering logics presented in a different way, say one with a Hilbert calculus and the other via a sequent calculus, we can define their fibring by fibring the induced consequence systems. We will also define other relations between consequence systems in Chapter 9.

1.2 Splicing and splitting

To start with, it is convenient to keep in mind that, in order to combine logics, we intend to depart from simple logics to obtain a more complex one. Following the terminology of [47], this process, by which a bunch of logics is synthesized forming a new logic, is called *splicing logics*. A prototypical case of splicing is the method of *fibring*, introduced in [104] (see also [108]). On the other hand, we may think about an analytic procedure that permits us to decompose a given logic into simpler components. This kind of process was called *splitting* in [47]. A prototypical case of splitting occurs when one succeeds in describing a given logic in terms of simpler components by means of translating the original logic into a collection of simpler, auxiliary logics, using what is called *possible-translations semantics*. This mechanism is introduced in [45] and further developed in [196] and [46].

We may thus consider two complementary approaches in the field of combining logic systems:

- Splicing, combining or composing logics: a bottom–up, synthetic approach presented in Figure 1.1. There are several methods to combine logics. Each method can be seen as an operation on some class of logics. In the figure, we start with logics $\mathcal{L}_1$ and $\mathcal{L}_2$ in some class of logics, and using the binary operation $\odot$ we obtain a new logic $\mathcal{L} = \mathcal{L}_1 \odot \mathcal{L}_2$.

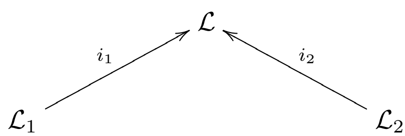


Figure 1.1: $\mathcal{L}$ is synthesized from $\mathcal{L}_1$ and $\mathcal{L}_2$

  The arrows i_1 and i_2 indicate that the component logics should be related with their combination. In general, these arrows induce consequence system morphisms from the component logics to their combination, meaning that derivation and entailment are preserved.

- Splitting, decombining or decomposing logics: a top–down, analytic approach presented in Figure 1.2. In this case we start with logic $\mathcal{L}$ and try to find an operation $\odot$ and components $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L} = \mathcal{L}_1 \odot \mathcal{L}_2$. The arrows f_1 and f_2 indicate that the given logic should be related with the components. In general, these arrows induce consequence system morphisms from the given logic to the components.

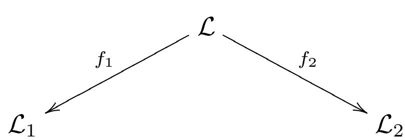


Figure 1.2: $\mathcal{L}$ is analyzed into $\mathcal{L}_1$ and $\mathcal{L}_2$

Splicing can be applied to more than two logics. It is worthwhile noting that splitting can involve an infinite number of components as we discuss below.

The known splicing mechanisms, as we will see throughout the book, have an important property. Once we choose a mechanism and the components $\mathcal{L}_1$ and $\mathcal{L}_2$, the resulting logic $\mathcal{L}$ is, in most cases, immediately defined. On the other hand, the known splitting mechanisms are different in this respect. Once we choose a mechanism and a logic $\mathcal{L}$, one can have several possibilities of choosing the components.

The splicing and the splitting mechanisms can be combined as we explain in Chapter 11 where we briefly discuss some applications. There we will discuss how one can use both of them if there is need to do so.

Among the most challenging problems in combination of logics we can refer to preservation of properties. For example, it is interesting to investigate if a logic resulting from a combination has a certain property, assuming that the original logics have that property. In many cases, one has to impose sufficient conditions for the preservation. The most relevant preservation results are related to soundness and completeness, interpolation and decidability.

Several combination mechanisms have been investigated, namely fusion of modal logics, product of modal logics, fibring by functions of modal logics, algebraic fibring, temporalization and parameterization, synchronization, institutions and parchments. They are targeted to different classes of logics and assume different degrees of abstraction. They all have in common that, in the end, we are dealing with consequence systems either with a syntactic or with a semantic nature. That is, in all the cases there are consequence operators that are extensive, monotonic and idempotent.

In this section, we only briefly describe three combination mechanisms: fusion, product and fibring by functions of modal logics. They will be used as examples in other chapters. For a complete description of fusion and product we refer to [113] and for a in depth treatment of fibring by functions see [108]. Algebraic fibring will be discussed in many guises throughout the book. The other combination mechanisms were triggered by applications in software specification and will be discussed in Chapter 11.

1.2.1 Fusion of modal logics

Fusion of normal modal logics (see [259]) is a binary operation on the class of normal modal logics endowed with Kripke semantics (introduced by Samuel Kripke in [174]) and Hilbert calculi, that we now explain with some detail. Recall that a Kripke structure is a triple

$$\langle W, R, V\rangle$$

where W is a non-empty set (the set of worlds), $R \subseteq W^2$ is a binary relation (the accessibility relation) and $V : \mathbb{P} \to \wp W$ is a map (the valuation).

Consider two normal modal logics $\mathcal{L}'$ and $\mathcal{L}''$ with the following characteristics:

- both have the same set $\mathbb{P}$ of zero-ary connectives (propositional constants), a unary connective $\neg$ and a binary connective $\Rightarrow$;
- a unary connective $\Box'$ and $\Box''$ for the logics $\mathcal{L}'$ and $\mathcal{L}''$, respectively;
- M' and M'' are classes of Kripke structures for $\mathcal{L}'$ and $\mathcal{L}''$, respectively;
- the Hilbert calculi H' and H'' include, besides the propositional part, the following axioms and rules:

– $\langle\emptyset, ((\Box'(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box'\xi_1) \Rightarrow (\Box'\xi_2)))\rangle$ K axiom for $\mathcal{L}'$;
– $\langle\emptyset, ((\Box''(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box''\xi_1) \Rightarrow (\Box''\xi_2)))\rangle$ K axiom for $\mathcal{L}''$;
– $\langle\{\xi\}, (\Box'\xi)\rangle$ necessitation rule for $\mathcal{L}'$;
– $\langle\{\xi\}, (\Box''\xi)\rangle$ necessitation rule for $\mathcal{L}''$.

The fusion of $\mathcal{L}'$ and $\mathcal{L}''$ is a normal bimodal logic $\mathcal{L}$ with two boxes that behave independently, except when otherwise imposed. That is, $\mathcal{L}$ is characterized as follows:

- a set $\mathbb{P}$ of zero-ary connectives (propositional constants), a unary connective $\neg$, a binary connective $\Rightarrow$ and two unary connectives $\Box'$ and $\Box''$;
- M is the class of all Kripke structures of the form $\langle W, R', R'', V\rangle$ where $\langle W, R', V\rangle$ and $\langle W, R'', V\rangle$ are Kripke structures of $\mathcal{L}'$ and $\mathcal{L}''$, respectively;
- the Hilbert calculus H includes all the rules of the Hilbert calculi of the original logics and hence the following ones:
  – $\langle\emptyset, ((\Box'(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box'\xi_1) \Rightarrow (\Box'\xi_2)))\rangle$ K axiom for $\mathcal{L}'$;
  – $\langle\emptyset, ((\Box''(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box''\xi_1) \Rightarrow (\Box''\xi_2)))\rangle$ K axiom for $\mathcal{L}''$;
  – $\langle\{\xi\}, (\Box'\xi)\rangle$ necessitation rule for $\mathcal{L}'$;
  – $\langle\{\xi\}, (\Box''\xi)\rangle$ necessitation rule for $\mathcal{L}''$.

Each model of the fusion corresponds to a model $\langle W, R', V'\rangle$ in $\mathcal{L}'$ and to a model $\langle W, R'', V''\rangle$ in $\mathcal{L}''$. That is, in the fusion we do not include models where the set W (of worlds) is different. In technical words, each model of the fusion should have as a reduct a model of $\mathcal{L}'$ and a model of $\mathcal{L}''$.

We briefly describe how a formula in the fusion is evaluated. Given the model $\langle W, R', R'', V\rangle$ in the fusion and $w \in W$ we have that the formula $(\Diamond'(\Box''p))$ is satisfied by $\langle W, R', R'', V\rangle$ at w, denoted by

$$\langle W, R', R'', V\rangle, w \Vdash (\Diamond'(\Box''p))$$

if there is $z \in W$ such that:

- $\langle W, R', R'', V\rangle, z \Vdash (\Box''p)$ and $wR'z$;
- $N_z \subseteq V(p)$ where $N_z = \{u \in W : zR''u\}$.

We refer to Figure 1.3 for details, where φ'_p is $(\Diamond'(\Box''p))$ and φ''_p is $(\Box''p)$.

Observe that the formula

$$(\Box''((\Box'(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box'\xi_1) \Rightarrow (\Box'\xi_2))))$$

is a theorem of the fusion. That is, we can derive this formulas from the K axiom for $\mathcal{L}'$ and the necessitation rule for $\mathcal{L}''$.

We synthesize the properties of fusion in the following way:

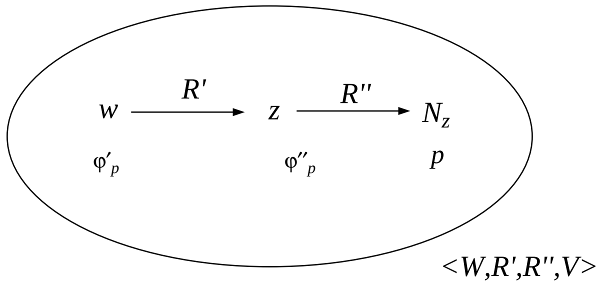


Figure 1.3: Evaluating the formula $(\Diamond'(\Box''p))$ in a fusion structure

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *homogeneous combination mechanism at the semantic level*: both original logics are presented by Kripke structures;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fusion;
- *algorithmic combination of logics at the semantic level*: given the classes of Kripke structures for the original logics, we know how to define the class of Kripke structures for the fusion.

The algorithmic nature of fusion also means that no interaction is stated between $\Box'$ and $\Box''$. We will see in Chapter 2 and in Chapter 3 that fusion is a canonical construction in the sense that it is minimal in some class of logics.

It is easy to conclude that the definition of $\mathcal{L}'$ above induces a consequence system

$$\mathcal{C}(H') = \langle L(C'), \vdash_{H'} \rangle$$

where C' is the signature of $\mathcal{L}'$ and $(\Gamma')^{\vdash_{H'}}$ is the set of formulas that can be derived from Γ', using the Hilbert calculus for $\mathcal{L}'$. In a similar way, we can define $\mathcal{C}''(H'') = \langle L(C''), \vdash_{H''} \rangle$ and $\mathcal{C}(H) = \langle L(C' \cup C''), \vdash_H \rangle$. Then we have:

$$\mathcal{C}(H') \leq \mathcal{C}(H) \text{ and } \mathcal{C}(H'') \leq \mathcal{C}(H).$$

The semantic characterizations of $\mathcal{L}'$, $\mathcal{L}''$ and $\mathcal{L}$ also induce consequence systems. Again, the consequence systems for $\mathcal{L}'$ and $\mathcal{L}''$ are weaker than the one for $\mathcal{L}$.

At first sight, this may seem a very simple combination mechanism. However, it is interesting enough for seeing that preservation of properties is not an easy issue. An example of a preservation problem can be presented in the following way. Assume that $\mathcal{L}'$ and $\mathcal{L}''$ are weakly complete logics with respect to the class of frames (that is, every valid formula is a theorem). Is the logic $\mathcal{L}$ resulting from the fusion also weakly complete with respect to the class of fusion frames? In fusion, there are preservation results for weak completeness, uniform Craig interpolation (for theoremhood) and decidability (see [281, 169]).

Fusion of non-normal modal logics was also investigated in [92], namely discussing preservation of weak completeness via a technique that extends the one used in the normal case.

It is worthwhile noting that there is no notion of fusion of a normal modal logic with a non-normal modal logic. Such a combination can, however, be defined in the context of algebraic fibring.

It is also worthwhile mentioning that the interested reader should also consult [99] where the notion of fusion was anticipated through some examples of combining alethic and deontic logics with philosophical interest.

1.2.2 Product of modal logics

We now concentrate on another mechanism of combination: the product of logics. The product of modal logics is a binary operation that is very useful when one wants, for example, to represent time-space information. Products were introduced in [235, 236]. We consider the same setting as we did for fusion of modal logics. The signature and the semantic counterparts of the product of $\mathcal{L}'$ and $\mathcal{L}''$ is as follows:

- a set $\mathbb{P}$ of zero-ary connectives (propositional constants), a unary connective $\neg$, a binary connective $\Rightarrow$ and two unary connectives $\Box'$ and $\Box''$;
- M is the class of product structures of the form

  $$\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V'' \rangle$$

  where $\langle W, R', V' \rangle$ and $\langle W, R'', V'' \rangle$ are Kripke structures of $\mathcal{L}'$ and $\mathcal{L}''$, respectively and where $\overline{R}', \overline{R}'' \subseteq (W' \times W'')^2$ are defined as follows:

  - $\langle w_1', w'' \rangle \overline{R}' \langle w_2' , w'' \rangle$ if $w_1' R' w_2'$;
  - $\langle w', w_1'' \rangle \overline{R}'' \langle w', w_2'' \rangle$ if $w_1'' R'' w_2''$;
  - $(V' \times V'')(p) = V'(p) \times V''(p)$.

The striking aspect about products is that some modal formulas are valid in every product frame. Namely the following will show that some interaction exists between $\Box'$ and $\Box''$ and $\Diamond'$ and $\Diamond''$ (recall that $\Diamond'\varphi$ is an abbreviation of $(\neg(\Box'(\neg \varphi)))$ and similarly with respect to $\Diamond''$):

- $((\Diamond'(\Diamond''p)) \Rightarrow (\Diamond''(\Diamond'p)))$ commutativity 1;
- $((\Diamond''(\Diamond'p)) \Rightarrow (\Diamond'(\Diamond''p)))$ commutativity 2;

- $((\Diamond'(\Box''p)) \Rightarrow (\Box''(\Diamond'p)))$ Church-Rosser property 1
- $((\Diamond''(\Box'p)) \Rightarrow (\Box'(\Diamond''p)))$ Church-Rosser property 2

For some time it was conjectured that product of Hilbert calculi was fusion of Hilbert calculi enriched with the four axioms above. For more details see [115, 116] where some counterexamples are analyzed. Hence, because of the interactions, it is not possible to obtain directly the Hilbert calculus for the product of two modal logics. The interaction axioms have to be fine tuned on the original logics.

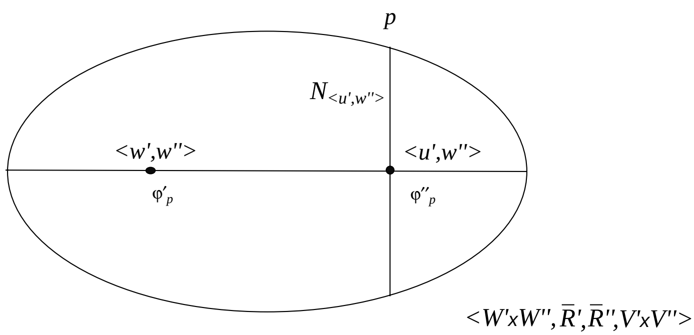


Figure 1.4: Evaluating the formula $(\Diamond'(\Box''p))$ in a product structure

We briefly describe how a formula in the product of modal logics is evaluated. Given the model $\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V''\rangle$ in the product and $\langle w', w''\rangle \in W$ we have that the formula $(\Diamond'(\Box''p))$ is satisfied by $\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V''\rangle$ at $\langle w', w''\rangle$, denoted by

$$\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V''\rangle, \langle w', w''\rangle \Vdash (\Diamond'(\Box''p))$$

if there is $\langle u', w''\rangle \in W' \times W''$ such that:

- $\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V''\rangle, \langle u', w''\rangle \Vdash (\Box''p)$ and $w'R'u'$;
- $N_{\langle u', w''\rangle} \subseteq (V' \times V'')(p)$ where $N_{\langle u', w''\rangle} = \{\langle u', u''\rangle \in W' \times W'' : w''R''u''\}$.

We refer to Figure 1.4 for details, where φ'_p is $(\Diamond'(\Box''p))$ and φ''_p is $(\Box''p)$.

We synthesize the properties of the product in the following way:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *homogeneous combination mechanism at the semantic level*: both original logics are presented by Kripke structures;

- *non algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we have to complete the definition of the Hilbert calculus of the product;

- *algorithmic combination of logics at the semantic level*: given the classes of Kripke structures for the original logics we know how to define the class of Kripke structures of the product.

It is easy to conclude that the definition of $\mathcal{L}'$ above induces a consequence system

$$\langle L(C'), \vDash' \rangle$$

where C' is the signature of $\mathcal{L}'$ and $(\Gamma')^{\vDash'}$ is the set of formulas that can be entailed from Γ', using the semantics for $\mathcal{L}'$. In a similar way, we can define $\langle L(C''), \vDash'' \rangle$ and $\langle L(C' \cup C''), \vDash \rangle$. Then we have:

$$\langle L(C'), \vDash' \rangle \leq \langle L(C' \cup C''), \vDash \rangle \text{ and } \langle L(C''), \vDash'' \rangle \leq \langle L(C' \cup C''), \vDash \rangle.$$

The above holds since:

- $\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V'' \rangle, \langle w', w'' \rangle \Vdash \varphi'$ if and only if $\langle W', R', V' \rangle, w' \Vdash \varphi'$;

- $\langle W' \times W'', \overline{R}', \overline{R}'', V' \times V'' \rangle, \langle w', w'' \rangle \Vdash \varphi''$
  if and only if $\langle W'', R'', V'' \rangle, w'' \Vdash \varphi''$.

Products of modal logics are useful to understand the semantics that we adopt in Chapter 6 for first-order based logics.

Several preservation and non preservation results were already obtained for products. Namely, in [120] several results are presented concerning products of transitive modal logics. Also in [199] some complexity results are presented.

1.2.3 Fibring by functions

In this section we briefly analyze the fibred semantics for modal logics, as originally presented in [104] (see also [108]). Herein, we give the name *fibring by functions* to this kind of combination. The reason will become clear below.

The setting is the same as the one for fusion of logics. Given a class M of Kripke structures, we denote by S_M the class of all pairs $\langle \langle W, R, V \rangle, w \rangle$ where $\langle W, R, V \rangle \in M$ and $w \in W$.

The fibring by functions of $\mathcal{L}'$ and $\mathcal{L}''$ is a normal bimodal logic $\mathcal{L}$ with two modal operators characterized as follows:

- the set $\mathbb{P}$ of zero-ary connectives (propositional constants), a unary connective $\neg$, a binary connective $\Rightarrow$ and unary connectives $\Box'$ and $\Box''$;

- M is a class of structures where each structure is either $\langle m', h', h'' \rangle$ or $\langle m'', h', h'' \rangle$ such that:

– $m' = \langle W', R', V' \rangle \in M'$ and $m'' = \langle W'', R'', V'' \rangle \in M''$, where M' and M'' are the classes of Kripke structures for $\mathcal{L}'$ and $\mathcal{L}''$, respectively;

– $h' : \biguplus_{m' \in M'} W' \to S_{M''}$ and $h'' : \biguplus_{m'' \in M''} W'' \to S_{M'}$ are maps.

A structure $\langle W', R', V', h', h'' \rangle$ *satisfies* a formula φ, written

$$\langle W', R', V', h' \rangle, w' \Vdash \psi$$

whenever the following conditions hold:

- $\langle m', h', h'' \rangle, w' \Vdash p$ if $w' \in V'(p)$;
- $\langle m', h', h'' \rangle, w' \Vdash (\neg \varphi)$ if $\langle m', h', h'' \rangle, w' \nVdash \varphi$;
- $\langle m', h', h'' \rangle, w' \Vdash (\varphi_1 \Rightarrow \varphi_2)$ if $\langle m', h', h'' \rangle, w' \nVdash \varphi_1$ or $\langle m', h', h'' \rangle, w' \Vdash \varphi_2$;
- $\langle m', h', h'' \rangle, w' \Vdash (\Box' \varphi)$ if $\langle m', h', h'' \rangle, v' \Vdash \varphi$ for every $v' \in W'$ such that $w' R' v'$;
- $\langle m', h', h'' \rangle, w' \Vdash (\Box'' \varphi)$ if $\langle m'', h', h'' \rangle, w'' \Vdash (\Box'' \varphi)$ where $h'(w') = \langle m'', w'' \rangle$.

The satisfaction when considering structures $\langle W'', R'', V'', h'' \rangle$ is defined in a similar way.

Let $(\Diamond' (\Box'' p))$ be a formula of the mixed language of $\mathcal{L}$. Assume that we want to evaluate this formula in the structure $\langle W', R', V', h', h'' \rangle$ at world w'. That is, we want to evaluate

$$\langle W', R', V', h', h'' \rangle, w' \Vdash (\Diamond' (\Box'' p)).$$

Since $(\Diamond' (\Box'' p)$ has $\Diamond'$ of the logic $\mathcal{L}'$ as an external modality, we can use the Kripke structure $\langle W', R', V' \rangle$. Thus, $\langle W', R', V', h', h'' \rangle, w' \Vdash (\Diamond' (\Box'' p))$ if there exists $v' \in W'$ such that $w' R' v'$ and $\langle W', R', V', h', h'' \rangle, v' \Vdash (\Box'' p)$. For the latter satisfaction, the Kripke structure $\langle W', R', V' \rangle$ is not enough. We have to use the

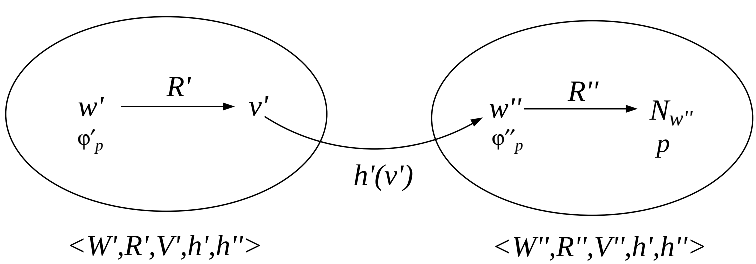


Figure 1.5: Evaluating the formula $(\Diamond' (\Box'' p))$ in fibring by functions structures

map h' that associates to v' a structure $\langle W'', R'', V''\rangle$ and a world $w'' \in W''$. That is, we have to evaluate

$$\langle W'', R'', V'', h', h''\rangle, w'' \Vdash (\Box'' p).$$

Finally, we have to evaluate p in all the neighbors of w'', that is, the elements of the set $\{v'' \in W'' : w'' R'' v''\}$.

The Figure 1.5, where φ'_p is $(\Diamond'(\Box'' p))$ and φ''_p is $(\Box'' p)$, illustrates this construction.

We synthesize the properties of fibring by functions in the following way:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *homogeneous combination mechanism at the semantic level*: both original logics are presented by Kripke structures;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the resulting Hilbert calculus;
- *semi-algorithmic combination of logics at the semantic level*: given the classes of Kripke structures for the original logics we know how to define the class of structures of the fibring by functions, provided that we are given the maps relating the models and worlds.

It is easy to conclude that the definition of $\mathcal{L}'$ above induces a consequence system

$$\langle L(C'), \vDash'\rangle$$

where C' is the signature of $\mathcal{L}'$ and $(\Gamma')^{\vDash'}$ is the set of formulas that can be entailed from Γ', using the semantics for $\mathcal{L}'$. In a similar way, we can define $\langle L(C''), \vDash''\rangle$ and $\langle L(C' \cup C''), \vDash\rangle$. Then we have:

$$\langle L(C'), \vDash'\rangle \leq_p \langle L(C' \cup C''), \vDash\rangle \text{ and } \langle L(C''), \vDash''\rangle \leq_p \langle L(C' \cup C''), \vDash\rangle.$$

The consequence systems $\langle L(C'), \vDash'\rangle$ and $\langle L(C''), \vDash''\rangle$ are weaker than the consequence system $\langle L(C' \cup C''), \vDash\rangle$ providing that we add some conditions on the maps h' and h''. We will discuss this issue later on in Section 3.4 of Chapter 3.

1.2.4 Gödel-Löb modal logic and Peano arithmetic

Perhaps the oldest decomposition of logics done in a rigorous context is related to the provability operator introduced by Kurt Gödel in 1933 (the original article is translated in [128]). For an historical account of the subject see [135] in [117]. For an overview of provability logic see also [156] in [33].

The idea is that we can translate provability logic GL into Peano arithmetic PA. We start by giving a brief overview of Peano arithmetic.

We follow [26] in the description of the Peano arithmetic below. The signature is a first-order signature with equality $\approx$, with function symbols $\mathbf{0}$ of arity 0, $\mathbf{s}$ of arity 1 and $+, \times$ of arity 2. We fix a set X of variables. Then the set of terms is as follows:

- x is a term for every $x \in X$;
- $\mathbf{0}$ is a term;
- $\mathbf{s}t, (t_1{+}t_2), (t_1{\times}t_2)$ are terms whenever t, t_1, t_2 are terms.

The set of formulas is defined as follows:

- $\perp, (t_1 \approx t_2)$ are formulas whenever t_1, t_2 are terms;
- $(\varphi_1 \Rightarrow \varphi_2), (\forall x \varphi)$ are formulas whenever $\varphi, \varphi_1, \varphi_2$ are formulas.

We denote by L_{PA} the language of PA. Negation $\neg$ and inequality $\not\approx$ are introduced as abbreviations of $(\varphi \Rightarrow \perp)$ and $(\neg(t_1 \approx t_2))$, respectively. We say that a sentence is a formula with no free variables. Besides the usual first-order axioms and rules we also have the following axioms in the Peano arithmetic (PA).

- $(\mathbf{0} \not\approx \mathbf{s}x)$;
- $((\mathbf{s}x \approx \mathbf{s}y) \Rightarrow (x \approx y))$;
- $((x{+}\mathbf{0}) \approx x)$;
- $((x{+}\mathbf{s}y) \approx \mathbf{s}(x{+}y))$;
- $((x{\times}\mathbf{0}) \approx \mathbf{0})$;
- $((x{\times}\mathbf{s}y) \approx ((x{\times}y){+}x))$;
- $(((\forall x((x \approx \mathbf{0}) \Rightarrow \varphi)) \wedge (\forall y((\forall x((x \approx y) \Rightarrow \varphi)) \Rightarrow (\forall x((x \approx \mathbf{s}y) \Rightarrow \varphi))))) \Rightarrow \varphi)$ where y does not occur in φ and is different from x.

We use $\vdash_{\text{PA}} \varphi$ to denote that φ is a theorem in Peano arithmetic.

Clearly the set of formulas is denumerable. Hence, we can establish a bijection between this set and the set of natural numbers. Choosing such a bijection g, the Gödel number of φ is $g(\varphi)$. Finite sequences of formulas can also be encoded in the natural numbers using g. Hence, given a sequence $\varphi_1 \dots \varphi_n$, $g(\varphi_1 \dots \varphi_n)$ denotes the Gödel number of the sequence. We denote by $\ulcorner \varphi \urcorner$ the numeral in PA for the Gödel number of the formula φ, that is, if n is the Gödel number of φ then $\ulcorner \varphi \urcorner$ is $\mathbf{0}$ preceded by n occurrences of the $\mathbf{s}$ operation.

In the sequel we denote by

$$Pf(y, x)$$

the formula stating that there is a proof (a finite sequence of formulas) with Gödel number y for the formula with Gödel number x. We also consider the formula

$$Bew(x)$$

as an abbreviation of $(\exists y Pf(y,x))$. Gödel established several properties of $Bew(\ulcorner\varphi\urcorner)$, namely the following:

- $\vdash_{\mathrm{PA}} (Bew(\ulcorner(\varphi_1 \Rightarrow \varphi_2)\urcorner) \Rightarrow (Bew(\ulcorner\varphi_1\urcorner) \Rightarrow Bew(\ulcorner\varphi_2\urcorner)))$;
- $\vdash_{\mathrm{PA}} (Bew(\ulcorner\varphi\urcorner) \Rightarrow Bew(\ulcorner Bew(\ulcorner\varphi\urcorner)\urcorner))$;
- if $\vdash_{\mathrm{PA}} \varphi$ then $\vdash_{\mathrm{PA}} Bew(\ulcorner\varphi\urcorner)$.

Clearly, there are similarities between Bew and a modal operator $\Box$.

Furthermore, Martin Hugo Löb [181], answering a question by Leon Henkin, proved the following property:

$$\text{if } \vdash_{\mathrm{PA}} (Bew(\ulcorner\varphi\urcorner) \Rightarrow \varphi) \text{ then } \vdash_{\mathrm{PA}} \varphi.$$

A modal logic $K4LR$ was introduced for studying modal properties of Bew seen as a modality $\Box$. This logic is a normal modal logic with the transitivity axiom 4, that is,

$$((\Box\varphi) \Rightarrow (\Box(\Box\varphi)))$$

and the rule

$$\langle\{(\Box\varphi \Rightarrow \varphi)\}, \varphi\rangle.$$

The operator $\Box$ in the logic $K4LR$ has all the required properties of Bew. However, it is more common to work with another modal logic for studying the properties of provability, the Gödel-Löb modal logic GL. This logic is a normal modal logic with a new axiom:

$$((\Box((\Box\varphi) \Rightarrow \varphi) \Rightarrow (\Box\varphi))).$$

These two modal logics are indeed the same. Let $\mathcal{C}(K4LR)$ and $\mathcal{C}(\mathrm{GL})$ be the consequence systems induced by the Hilbert calculi for the logics $K4LR$ and GL, respectively. Then

$$\mathcal{C}(K4LR) \leq_p \mathcal{C}(\mathrm{GL}) \;\text{ and }\; \mathcal{C}(\mathrm{GL}) \leq_p \mathcal{C}(K4LR)$$

that is the GL-theorems are the same as the $K4LR$-theorems.

The logic GL can be seen as splitting through PA. For this purpose we need to introduce the notion of realization. We denote by L_{GL} the language of the logic GL which we assume to be generated by a set of propositional constants $\mathbb{P}$.

A *realization* is a map $\bar{\lambda} : \mathbb{P} \to L_{\mathrm{PA}}$ that assigns to each $p \in \mathbb{P}$ a sentence. Each realization induces a map

$$\lambda : L_{\mathrm{GL}} \to L_{\mathrm{PA}}$$

such that:

- $\lambda(p) = \bar{\lambda}(p)$;
- $\lambda(\bot) = \bot$;
- $\lambda(\varphi \Rightarrow \psi) = (\lambda(\varphi) \Rightarrow \lambda(\psi))$;
- $\lambda(\Box\varphi) = Bew(\lceil \lambda(\varphi) \rceil)$.

This map is called a translation of $L_{\rm GL}$ into $L_{\rm PA}$. We denote by Λ the set of all such translations.

The relationship between GL and PA is as follows (see [252]):

$$\vdash_{\rm GL} \varphi \ \text{ if and only if } \ \vdash_{\rm PA} \lambda(\varphi) \text{ for every } \lambda \in \Lambda$$

where, as expected, $\vdash_{\rm GL} \varphi$ states that φ is a GL-theorem. Hence, if φ is a theorem of GL then all its possible translations (via realizations) into PA are also theorems of PA and vice-versa.

For some developments related to fusion of provability logics see also [156].

In Example 1.4.5 below we shall see that the representation of provability logic by means of translations into Peano arithmetic is an instance of the splitting technique called *possible-translations characterizations*, to be analyzed in Chapter 9.

1.3 Algebraic fibring

Herein, we present fibring in a very simple context also with the intention of motivating that it is a universal or canonical construction. We use the name algebraic fibring because as referred in [1] we investigate the algebraic essence of the constructions and use algebraic constructions following the tradition of algebraic logic started by George Boole and later on by Augustus De Morgan [81].

The first observation is that we want to extend the constructions already described in Section 1.2 to the situation in which we can have logics that are not necessarily modal. Then, fusion will come as a particular case.

The second observation has to do with the specifics of the classes of logics that we assume to have as original logics. In this section, we consider that the target logics are propositional based (they do not involve any quantifier operators) and that we are in a homogeneous scenario. That is, both logics are presented deductively in the same way.

Hence, the construction has the following properties:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fibring.

The starting point is to fix the signatures. We assume that the original logics have a propositional signature as the one described in Definition 1.1.11. We also assume a denumerable set $\Xi = \{\xi_i : i \in \mathbb{N}\}$ of schema variables that will be useful when defining a general notion of Hilbert calculus.

In order to define fibring of signatures we need to be able to compare signatures. Recall that $C \leq C'$ means $C_k \subseteq C'_k$ for every $k \in \mathbb{N}$.

The fibring of two signatures C' and C'' is the union $C' \cup C''$. Hence $C' \leq C' \cup C''$ and $C'' \leq C' \cup C''$.

We can give this definition in a more general way by using morphisms:

Definition 1.3.1 Let C and C' be propositional signatures. A *signature morphism* $h : C \to C'$ is a family of maps $h_k : C_k \to C'_k$, for every $k \in \mathbb{N}$. ∇

The advantage of using morphisms is that we can use the names we want for the different symbols in the signatures at hand, translating them between signatures using the morphisms. We assume that a connective in one of the signatures corresponds to a connective of the same arity in the other signature. Signatures and their morphisms constitute the category **Sig**.

In this category, fibring of signatures is a universal construction. When there is no sharing of connectives then fibring is a coproduct in **Sig**. This means that (see Figure 1.6):

- $C' \cup C''$ is a disjoint union of C' and C'', that is, the disjoint union of C'_k and C''_k for each $k \in \mathbb{N}$;
- there are morphisms $i' : C' \to C' \cup C''$ and $i'' : C'' \to C' \cup C''$;
- given signature morphisms $h' : C' \to C$ and $h'' : C'' \to C$, there is a unique signature morphism $h : C' \cup C'' \to C$ such that $h \circ i' = h'$ and $h \circ i'' = h''$.

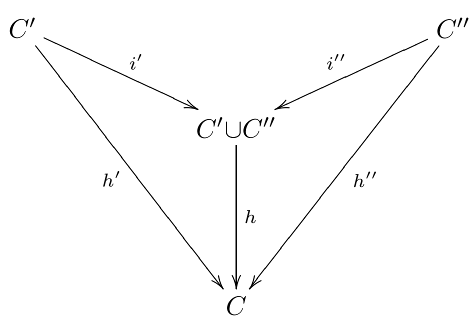


Figure 1.6: Coproduct of signatures

Therefore, unconstrained fibring (fibring with no sharing) is the minimal signature in the class of all signatures that include C' and C'' and do not identify any pair of connectives from C' and C''. The last condition of the coproduct ensures the minimality.

Example 1.3.2 Assume that we want to define the unconstrained fibring of the signatures:

- C° such that $C_1^\circ = \{\neg, \circ\}$ and $C_2^\circ = \{\wedge, \vee, \Rightarrow\}$;
- $C^\Box$ such that $C_1^\Box = \{\neg, \Box\}$ and $C_2^\Box = \{\Rightarrow\}$.

Their unconstrained fibring is depicted in Figure 1.7.

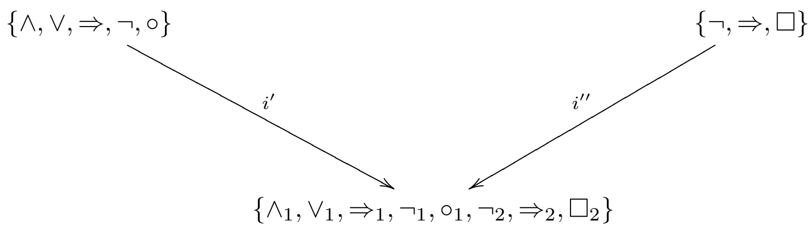


Figure 1.7: Example of unconstrained fibring of signatures

For simplicity, we sometimes join together in just one set all the connectives independently of their arities. Observe that in the fibring the connectives have a different name so that no mixture arises. Note that another choice for the unconstrained fibring signature is as follows:

$$\{\wedge, \vee, \Rightarrow_1, \neg_1, \circ, \neg_2, \Rightarrow_2, \Box\}$$

where we only change the names of the connectives that are present in both signatures. ∇

When we want to share symbols in C' and C'' we must start by identifying the common symbols, by defining a new signature $\overline{C}$ and the signature morphisms from $\overline{C}$ to C' and C'' as in Figure 1.8.

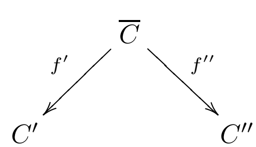


Figure 1.8: Shared signature

Afterward, we define the fibring $C' \cup C''$ in the following way (see Figure 1.9):

- define the coproduct $C' \oplus C''$ of C' and C'' with $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$;
- calculate the coequalizer $\langle C' \cup C'', q\rangle$ of $i' \circ f'$ and $i'' \circ f''$ where:
  - $C' \cup C'' = i'(C' \setminus f'(\overline{C})) \cup i''(C'' \setminus f''(\overline{C})) \cup \overline{C}$;
  - $q : C' \oplus C'' \to C' \cup C''$ is a signature morphism, such that:

$$q(i'(c')) = \begin{cases} c & \text{if } c' \text{ is } f'(c) \text{ for some } c \in \overline{C} \\ i'(c') & \text{otherwise;} \end{cases} ;$$

$$q(i''(c'')) = \begin{cases} c & \text{if } c'' \text{ is } f''(c) \text{ for some } c \in \overline{C} \\ i''(c'') & \text{otherwise;} \end{cases} .$$

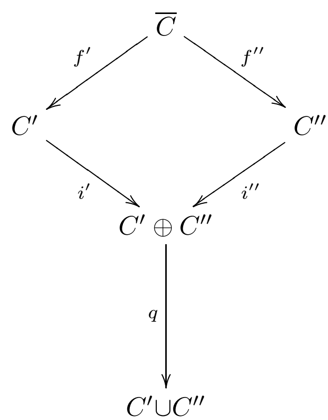


Figure 1.9: Pushout of signatures

Observe that in the resulting fibring signature $C' \cup C''$ we just have the common symbols identified in $\overline{C}$ plus the non shared symbols from C' and C'', possibly with a different name to avoid mixing them. Technically, we say that the constrained fibring $C' \cup C''$ is a pushout of C' and C'' together with the morphisms $f' : \bar{C} \to C'$ and $f'' : \bar{C} \to C''$.

Example 1.3.3 Assume we want to define the constrained fibring of signatures:

- C' such that $C'_1 = \{\neg, \circ\}$ and $C'_2 = \{\sqcap, \sqcup, \to\}$;
- C'' such that $C''_1 = \{\sim, \Box\}$ and $C''_2 = \{\cap, \cup, \supset\}$;

sharing the connectives of arity 2. Then, we can consider the shared signature $\overline{C}$ as follows (see Figure 1.10):

- $\overline{C}_2 = \{\wedge, \vee, \Rightarrow\}$;
- $f'(\wedge) = \sqcap$, $f'(\vee) = \sqcup$, $f'(\Rightarrow) = \rightarrow$;
- $f''(\wedge) = \cap$, $f'(\vee) = \cup$, $f'(\Rightarrow) = \supset$.

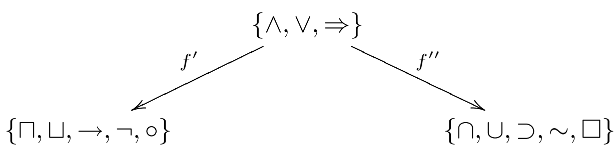


Figure 1.10: Example of sharing of a signature

We detail the construction of the constrained fibring in Figure 1.11.
In the first step, we define the coproduct $C' \oplus C''$ obtaining:

$$\{\sqcap_1, \sqcup_1, \rightarrow_1, \neg_1, \circ_1, \cap_2, \cup_2, \supset_2, \sim_2, \Box_2\}$$

and where the morphisms i' and i'' are such that:

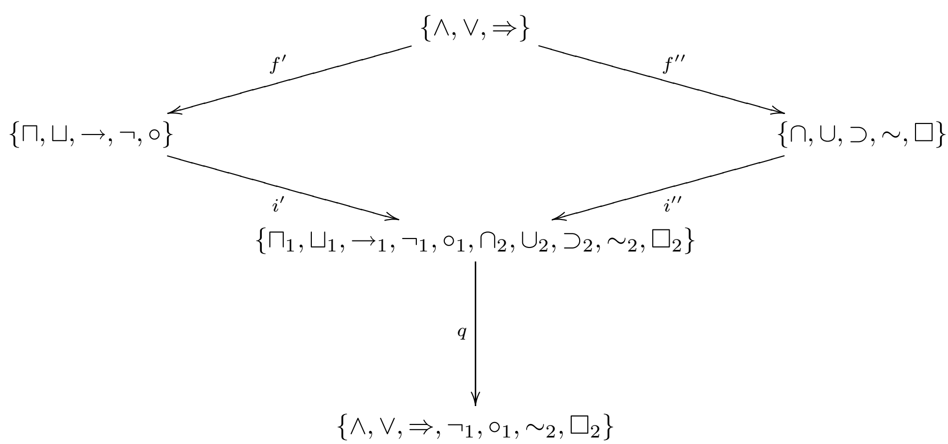


Figure 1.11: Example of a constrained fibring signature

- $i'(\sqcap) = \sqcap_1$, $i'(\sqcup) = \sqcup_1$ $i'(\rightarrow) = \rightarrow_1$, $i'(\neg) = \neg_1$ and $i'(\circ) = \circ_1$;
- $i''(\cap) = \cap_2$, $i''(\cup) = \cup_2$, $i''(\supset) = \supset_2$, $i''(\sim) = \sim_2$ and $i'(\Box) = \Box_2$.

Then we calculate the coequalizer of $i' \circ f'$ and $i'' \circ f''$ and obtain the signature

$$\{\wedge, \vee, \Rightarrow, \neg_1, \circ_1, \sim_2, \Box_2\}$$

and the signature morphism q that identifies in the coproduct signature the symbols to be shared. Consider for instance the connective $\Rightarrow$ in the fibring. This connective corresponds to the sharing of connectives $\rightarrow$ and $\supset$.

It could also be possible to define the constrained fibring as $\{\wedge, \vee, \Rightarrow, \neg, \circ, \sim, \Box\}$. ∇

It is worthwhile to refer that constrained fibring is again a minimal construction. That is, $C' \cup C''$ is the minimal signature among those for which there are signature morphisms $g' : C' \to C$ and $g'' : C'' \to C$ such that

$$g' \circ f' = g'' \circ f''.$$

Indeed there is a unique signature morphism $h : C' \cup C'' \to C$ such that

$$h \circ q \circ i' = g' \text{ and } h \circ q \circ i'' = g''.$$

This means that $C' \cup C''$ is the minimal signature among those that include the shared signature plus the non shared symbols of both signatures C' and C'', possibly with different names.

Recall from Definition 1.1.12, that each signature induces a language $L(C)$. A signature morphism h induces a unique translation $\hat{h} : L(C) \to L(C')$ between the corresponding languages as follows:

- $\hat{h}(\xi) = \xi$;
- $\hat{h}(c(\varphi_1, \ldots, \varphi_k)) = h_k(c)(\hat{h}(\varphi_1), \ldots, \hat{h}(\varphi_k))$.

In the sequel, we will often use $h(\varphi)$ instead of $\hat{h}(\varphi)$.

We assume that the original logics are presented as Hilbert calculi.

Fibring Hilbert calculi is just putting together their signatures and inference rules. Hence the fibring of $H' = \langle C', R' \rangle$ and $H'' = \langle C'', R'' \rangle$, denoted by $H' \cup H''$ is the Hilbert calculus

$$\langle C' \cup C'', R' \cup R'' \rangle.$$

We say that H is *weaker* than H', denoted by

$$H \leq H'$$

when $C \leq C'$ and $R \subseteq R'$. In particular, $H' \leq H' \cup H''$ and $H'' \leq H' \cup H''$. As a consequence, every derivation in H' is also a derivation in $H' \cup H''$. The same comment can be applied to derivations in H''.

Hilbert calculi can also be related by morphisms. There are two natural possibilities. The following is the more usual one.

Definition 1.3.4 A *Hilbert calculus morphism* $h : H \to H'$ is a signature morphism $h : C \to C'$ such that $h(\Gamma) \vdash_{H'} h(\varphi)$ when $\Gamma \vdash_H \varphi$. ∇

According to this definition there is a morphism between Hilbert calculi when there is a morphism between the corresponding induced consequence systems. The other possibility is to require that the translation of each rule of H is also a rule in H'. This approach is adopted in Chapters 2 and 5. Of course, any morphism in this sense is also a morphism in the sense of Definition 1.3.4.

An alternative characterization of morphism in the sense of Definition 1.3.4 is sometimes useful.

Proposition 1.3.5 *A signature morphism $h : C \to C'$ is a Hilbert calculus morphism $h : \langle C, R\rangle \to \langle C', R'\rangle$ if and only if $h(\Delta) \vdash_{\langle C',R'\rangle} h(\varphi)$ for every inference rule $\langle \Delta, \varphi\rangle \in R$.* ∇

Hilbert calculi and their morphisms constitute the category **Hil**. We now characterize fibring as a universal construction in this category.

When there is no sharing of connectives then the fibring

$$H' \cup H'' = \langle C' \cup C'', R\rangle$$

of Hilbert calculi H' and H'' is a coproduct in **Hil**. This means that (see Figure 1.12):

- $C' \cup C''$ is the unconstrained fibring of signatures C' and C'' with signature morphisms $i' : C' \to C' \cup C''$ and $i'' : C'' \to C' \cup C''$;
- $R = \{i'(r) \ : \ r \in R'\} \cup \{i''(r) \ : \ r \in R''\}$;
- the signature morphisms i' and i'' induce the following Hilbert calculus morphisms $i' : H' \to H' \cup H''$ and $i'' : H'' \to H' \cup H''$;
- given Hilbert calculus morphisms $h' : H' \to H$ and $h'' : H'' \to H$, there is a unique Hilbert calculus morphism $h : H' \cup H'' \to H$ such that $h \circ i' = h'$ and $h \circ i'' = h''$.

Therefore, in the unconstrained fibring $H' \cup H''$ we have all the symbols in the signatures of the original Hilbert calculi, as well as all the inference rules, possibly with some changes in the names of the connectives.

Example 1.3.6 Consider again the signatures C° and $C^\Box$ as introduced in Example 1.3.2. Assume that the Hilbert calculi $H^\circ = \langle C^\circ, R^\circ\rangle$ and $H^\Box = \langle C^\Box, R^\Box\rangle$ are such that:

- R° includes:
  - $\langle \emptyset, ((\neg(\circ\xi_1)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_1)))\rangle$;
  - $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$.
- $R^\Box$ includes:

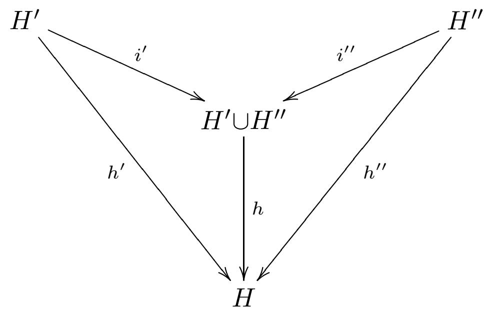


Figure 1.12: Coproduct of Hilbert calculi

– $\langle\emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2)))\rangle$;
– $\langle\{\xi_1\}, (\Box\xi_1)\rangle$;
– $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$.

The unconstrained fibring of H° and $H^\Box$ is then

$$H^\circ \cup H^\Box = \langle C^\circ \cup C^\Box, R\rangle$$

where $R = i'(R^\circ) \cup i''(R^\Box)$ and i' and i'' are as in Example 1.3.2. That is, R includes for instance:

- $\langle\emptyset, ((\neg_1(\circ_1\xi_1)) \Rightarrow_1 (\xi_1 \wedge_1 (\neg_1 \xi_1)))\rangle$;
- $\langle\{\xi_1, (\xi_1 \Rightarrow_1 \xi_2)\}, \xi_2\rangle$;
- $\langle\emptyset, ((\Box_2(\xi_1 \Rightarrow_2 \xi_2)) \Rightarrow_2 ((\Box_2\xi_1) \Rightarrow_2 (\Box_2\xi_2)))\rangle$;
- $\langle\{\xi_1\}, (\Box_2\xi_1)\rangle$;
- $\langle\{\xi_1, (\xi_1 \Rightarrow_2 \xi_2)\}, \xi_2\rangle$.

Observe that the two rules $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$ are not to be confused in the fibring. ∇

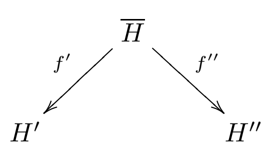


Figure 1.13: Sharing Hilbert calculi

We now turn our attention to the case of constrained fibring of Hilbert calculi. When we want to share symbols in C' and C'' we must start by identifying the common symbols, by defining a new Hilbert calculus $\overline{H}$ and the Hilbert calculus morphisms from $\overline{H}$ to both H' and H'', as in Figure 1.13. The appropriate Hilbert calculus for this purpose is $\overline{H} = \langle \overline{C}, \emptyset \rangle$ where $\overline{C}$ is the shared signature. Hence, for sharing, we only have to worry about the signatures.

Next, we define the fibring $H' \cup H''$ as a pushout in the following way (see also Figure 1.14):

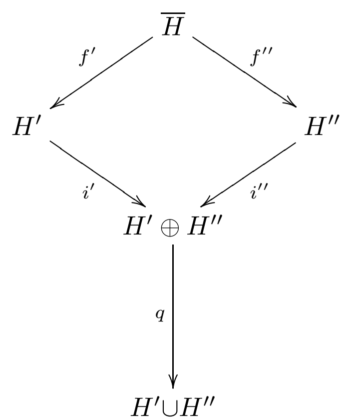


Figure 1.14: Pushout of Hilbert calculi

- define the coproduct $H' \oplus H''$ of H' and H'' with i' and i'' as above;
- calculate the coequalizer $\langle H' \cup H'', q \rangle$ of $i' \circ f'$ and $i'' \circ f''$:

$$H' \cup H'' = \langle C' \cup C'', q(i'(R')) \cup q(i''(R'')) \rangle$$

  where:

  - $C' \cup C''$ is the constrained fibring of signatures C' and C'';
  - $q : H' \oplus H'' \to H' \cup H''$ is the Hilbert calculus morphism corresponding to the coequalizer of the signature morphisms $i' \circ f'$ and $i'' \circ f''$.

Example 1.3.7 Consider again Example 1.3.6. Assume that we want to share $\Rightarrow$. The constrained fibring signature is as in Figure 1.15. As a result the constrained fibring of Hilbert calculi H° and $H^\square$ sharing $\Rightarrow$ is as follows:

$$H^\circ \cup H^\square = \langle C^\circ \cup C^\square, R \rangle$$

where:

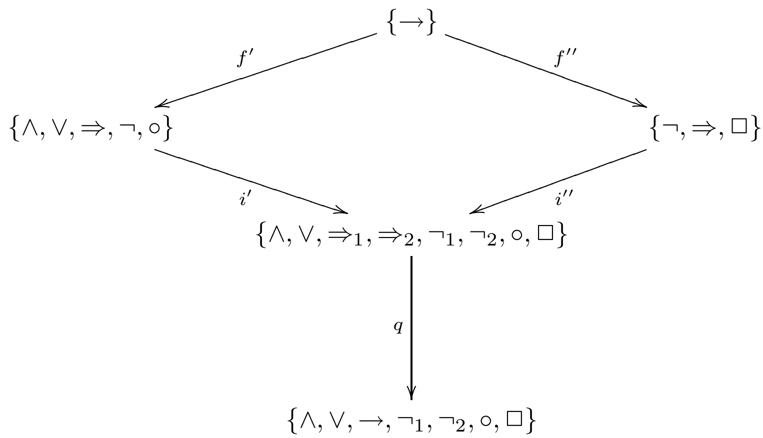


Figure 1.15: Example of a constrained fibring signature

- $C^{\circ} \cup C^{\Box}$ is as in Figure 1.15;
- R includes the following inference rules:
  - $\langle \emptyset, ((\neg_1(\circ\xi_1)) \to (\xi_1 \wedge (\neg_1\, \xi_1))) \rangle$;
  - $\langle \{\xi_1, (\xi_1 \to \xi_2)\}, \xi_2 \rangle$.
  - $\langle \emptyset, ((\Box(\xi_1 \to \xi_2)) \to ((\Box\xi_1) \to (\Box\xi_2))) \rangle$;
  - $\langle \{\xi_1\}, (\Box\xi_1) \rangle$.

Observe that the two modus ponens rules $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$ collapsed. All the original rules admit new instances. For instance, we can substitute ξ_1 by $(\circ\xi_1)$ and so derive $(\Box(\circ\xi_1))$ from $(\circ\xi_1)$ by using the rule $\langle \{\xi_1\}, (\Box\xi_1) \rangle$. This is a new instance of the rule, since originally the connective $\circ$ does not belong to the signature of $H^{\Box}$.

∇

Sometimes it is more useful to obtain the unconstrained fibring as the codomain of a cocartesian lifting (see [15] for a gentle introduction and [154] for more advanced aspects) taking advantage of forgetful functors into the category of signatures. That is the case in Chapter 7.

It is worthwhile to see the relationship between the consequence system generated by the fibring and the consequence systems generated by the components. We do so in the case of constrained fibring. The case of unconstrained fibring is analogous. First observe that $q \circ i'$ and $q \circ i''$ translate formulas in $L(C')$ and formulas of $L(C'')$ into formulas in $L(C' \cup C'')$. Then we have

$$(q \circ i')(\mathcal{C}(H')) \cup (q \circ i'')(\mathcal{C}(H'')) \leq \mathcal{C}(H' \cup H'')$$

but not always

$$\mathcal{C}(H' \cup H'') \leq (q \circ i')(\mathcal{C}(H')) \cup (q \circ i'')(\mathcal{C}(H'')).$$

Example 1.3.8 Consider Example 1.3.6. Let $\mathcal{C}(H' \cup H'') = \langle C' \cup C'', \mathsf{C} \rangle$, $\mathcal{C}(H') = \langle C', \mathsf{C}' \rangle$ and $\mathcal{C}(H'') = \langle C'', \mathsf{C}'' \rangle$. Then,

- $(\Box(\circ\xi)) \in \mathsf{C}(\{(\circ\xi)\})$;
- but not $(\Box(\circ\xi)) \in (q \circ i')(\mathsf{C}'(\{(\circ\xi)\})) \cup (q \circ i'')(\mathsf{C}''(\{(\circ\xi)\}))$. ∇

The fibring construction briefly illustrated above describes the main ideas for more complicated fibring constructions that we discuss throughout the book. Indeed, most of the fibring constructions that we consider in other chapters are all proved to be universal, either minimal or maximal. Moreover, we also show in Chapter 3 that fusion is a minimal construction. The same applies to fibring of functions providing that some conditions are added.

In Chapter 2 we discuss fibring of Hilbert calculi in more detail. Namely, we discuss some preservation results for some metatheorems and interpolation. In Chapter 3 we give a semantic account of algebraic fibring namely analyzing preservation of soundness and completeness. In both chapters only propositional based logics are considered. In Chapter 6, fibring is discussed for first-order based logics. In Chapter 7, fibring is extended to the context of higher-order logics.

1.4 Possible-translations semantics

This section intends to give a brief overview of a splitting mechanism: the method of possible-translations semantics (see [45]). This method was introduced to help solving the problem of assigning semantic interpretations to non-classical logics, for instance to non-truth functional logics.

The basic idea, if we want to decompose a given logic in terms of others by means of translations, is to see it as a ciphered text that we want to decode completely. If we translate the text into a single language, or into a bunch of other known languages, such that we have some guarantee to have covered all the subtleties of the ciphered text, then we know we have grasped the encrypted meaning.

We start by defining the central notion of translation between logics. We observe that we may need more than a single translation.

Definition 1.4.1 Let $\mathcal{C}_i = \langle C^i, \mathsf{C}_i \rangle$, $i = 1, 2$, be two consequence systems, and let $f : L(C^1) \to L(C^2)$ be a map.

- f is said to be a *weak translation* between $\mathcal{C}_1$ and $\mathcal{C}_2$ if $f(\mathsf{C}_1(\emptyset)) \subseteq \mathsf{C}_2(f(\emptyset))$;
- f is said to be a *translation* between $\mathcal{C}_1$ and $\mathcal{C}_2$ if $f(\mathsf{C}_1(\Gamma)) \subseteq \mathsf{C}_2(f(\Gamma))$, for every $\Gamma \subseteq L(C^1)$;

- f is a *conservative translation* between $\mathcal{C}_1$ and $\mathcal{C}_2$ if $f(\mathsf{C}_1(\Gamma)) = \mathsf{C}_2(f(\Gamma))$, for every $\Gamma \subseteq L(C^1)$. ▽

Observe that a translation f is a map between the languages not necessarily induced by a map between the underlying signatures. If the map is induced by a signature morphism then a translation is a consequence system morphism.

Example 1.4.2 Let $\mathcal{C} = \langle L, \models \rangle$ be the consequence system corresponding to classical propositional logic where L is generated by a set $\mathbb{P}$ of propositional constants and $\models$ is the semantic entailment. Let also $\mathcal{C}' = \langle L', \models' \rangle$ be the consequence system corresponding to a modal propositional logic and $\models'$ is the semantic entailment induced by Kripke structures.

Consider a map $\bar{f} : \mathbb{P} \to L'$ and its extension $f : L \to L'$ defined as follows:

- $f(p) = \bar{f}(p)$;
- $f(\neg\, \varphi) = (\neg\, f(\varphi))$;
- $f(\varphi \Rightarrow \psi) = (f(\varphi) \Rightarrow f(\psi))$.

We are going to prove that f is a weak translation between $\mathcal{C}$ and $\mathcal{C}'$.

Each Kripke structure $m = \langle W, R, V \rangle$ and $w \in W$ induce a valuation

$$v_{mw} : \mathbb{P} \to \{0, 1\}$$

as follows: $v_{mw}(p) = 1$ if $m, w \Vdash' f(p)$ for every $p \in \mathbb{P}$.

It is easy to prove that

$$v_{mw} \Vdash \varphi \text{ if and only if } m, w \Vdash f(\varphi).$$

Using this fact we can easily prove that if $\varphi \in \emptyset^{\models}$ then $f(\varphi) \in \emptyset^{\models'}$. Hence we can conclude that f is a weak translation.

As a consequence of this, we can say that every classical tautology is translated by f into a tautological modal formula. ▽

Definition 1.4.3 Let $\mathcal{C} = \langle C, \mathsf{C} \rangle$ be a consequence system, and let $\{\mathcal{C}_i\}_{i \in I}$ be a family of consequence systems such that $\mathcal{C}_i = \langle C^i, \mathsf{C}_i \rangle$ for every $i \in I$. A *possible-translations frame* for $\mathcal{C}$ is a pair

$$P = \langle \{\mathcal{C}_i\}_{i \in I}, \{f_i\}_{i \in I} \rangle$$

such that $f_i : L(C) \to L(C^i)$ is a translation between $\mathcal{C}$ and $\mathcal{C}_i$, for every $i \in I$. We say that $P = \langle \{\mathcal{C}_i\}_{i \in I}, \{f_i\}_{i \in I} \rangle$ is a *possible-translations characterization for* $\mathcal{C}$ if, for every $\Gamma \cup \{\varphi\} \subseteq L(C)$,

$$\varphi \in \mathsf{C}(\Gamma) \text{ if and only if } f_i(\varphi) \in \mathsf{C}_i(f_i(\Gamma)) \text{ for every } i \in I.$$

When C is of semantic nature we say that P is a *possible-translations semantics* for $\mathcal{C}$. ▽

Note that in a possible-translations characterization $P = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$ for $\mathcal{C}$ each map f_i is a consequence system morphism.

When $I = \{1, \ldots, n\}$, to test whether $\varphi \in \mathsf{C}(\Gamma)$ amounts to performing n tests: $f_i(\varphi) \in \mathsf{C}_i(f_i(\Gamma))$ for $1 \leq i \leq n$.

A possible-translations characterization for a given consequence system $\mathcal{C}$ can be regarded as a way to decompose $\mathcal{C}$ into the family $\{\mathcal{C}_i\}_{i\in I}$ through the translations $\{f_i\}_{i\in I}$.

A weaker notion of possible-translations characterization can be considered, by using weak translations. Thus, a *weak possible-translations frame* is a possible-translations frame in which the mappings are weak translations. And a *weak possible-translations characterization* for $\mathcal{C}$ is a weak possible-translations frame which only characterizes theoremhood of $\mathcal{C}$, that is, for every $\Gamma \cup \{\varphi\} \subseteq L(C)$,

$$\varphi \in \mathsf{C}(\emptyset) \text{ if and only if } f_i(\varphi) \in \mathsf{C}_i(f_i(\emptyset)) \text{ for every } i \in I.$$

In the following we analyze some examples.

Example 1.4.4 Consider again Example 1.4.2. Assume that the consequence system for modal logic is also generated by the set $\mathbb{P}$. Take F as the set of maps $f : L(\mathbb{P}) \to L'$ generated by all maps $\bar{f} : \mathbb{P} \to L'$ introduced in Example 1.4.2. Then $\langle\{\mathcal{C}_f\}_{f\in F}, F\rangle$, where $\mathcal{C}_f$ is $\mathcal{C}'$ for every $f \in F$, is a weak possible-translations semantics for $\mathcal{C}$. $\triangledown$

Example 1.4.5 Recall that, in Subsection 1.2.4, we have shown that the logic of provability GL can be seen as splitting through Peano Arithmetic PA. That characterization can be recast in terms of possible–translations.

Let $\mathcal{C}(\mathrm{PA})$ be the consequence system induced by PA. Let $\mathcal{C}_\lambda = \mathcal{C}(\mathrm{PA})$ for every translation $\lambda : L_{\mathrm{GL}} \to L_{\mathrm{PA}}$ induced by a realization $\bar{\lambda} : \mathbb{P} \to L_{\mathrm{PA}}$. Then

$$\langle\{\mathcal{C}_\lambda\}_{\lambda\in\Lambda}, \Lambda\rangle$$

is a weak possible-translations characterization for $\mathcal{C}(\mathrm{GL})$. $\triangledown$

Example 1.4.6 The *logics of formal inconsistency* (**LFI**s) were introduced in [51] (see also [49]). The **LFI**s are paraconsistent logics in which the notions of consistency and/or inconsistency are internalized at the object–language level. This is done by means of (primitive or defined) unary connectives $\bullet$ for inconsistency and/or $\circ$ for consistency, which satisfy suitable axioms.

For instance, the well-known paraconsistent logic $\mathfrak{C}_1$ of da Costa (see Example 2.2.9 of Chapter 2) is an **LFI**. Other interesting examples of **LFI**s are the logics **bC** and **Ci** , and its weaker versions **mCi** and **mbC** (see [51, 49]). The **LFI**s will be treated again in Chapters 5 and 9. In this example we will give a possible-translations semantics for **Ci**.

A signature C° for **Ci** is such that

- $C_1^\circ = \{\neg, \circ\}$;
- $C_2^\circ = \{\vee, \wedge, \Rightarrow\}$;
- $C_k^\circ = \emptyset$ in any other case.

The consequence system $\mathcal{C}_{\mathbf{Ci}} = \langle C^\circ, \vDash_{\mathbf{Ci}}\rangle$ for the logic **Ci** (see [49]) can be presented by the class of all maps (called *bivaluations*)

$$v : L(C^\circ) \to \{0, 1\}$$

where 1 is the designated value, such that

(v1) $v(\varphi \wedge \psi) = 1$ if and only if $v(\varphi) = 1$ and $v(\psi) = 1$;

(v2) $v(\varphi \vee \psi) = 1$ if and only if $v(\varphi) = 1$ or $v(\psi) = 1$;

(v3) $v(\varphi \Rightarrow \psi) = 1$ if and only if $v(\varphi) = 0$ or $v(\psi) = 1$;

(v4) if $v(\neg\varphi) = 0$ then $v(\varphi) = 1$;

(v5) if $v(\neg(\neg\varphi)) = 1$ then $v(\varphi) = 1$;

(v6) if $v(\circ\varphi) = 1$ then $v(\varphi) = 0$ or $v(\neg\varphi) = 0$;

(v7) if $v(\neg(\circ\varphi)) = 1$ then $v(\varphi) = 1$ and $v(\neg\varphi) = 1$.

Observe that $\Gamma \vDash_{\mathbf{Ci}} \varphi$ if, for every bivaluation v, $v(\varphi) = 1$ whenever $v(\gamma) = 1$ for every $\gamma \in \Gamma$.

We now describe the consequence system into which $\mathcal{C}_{\mathbf{Ci}}$ is to be translated. It is defined over the signature C such that:

- $C_1 = \{\neg_1, \neg_2, \circ_1, \circ_2\}$;
- $C_2 = \{\vee, \wedge, \Rightarrow\}$;
- $C_k = \emptyset$ in any other case.

Consider the matrix, that is, the algebra, M_0 for the signature C where the set of truth-values is $\{T, t, F\}$ with $D = \{T, t\}$ as the set of designated values and with the truth-tables displayed below.

$\wedge$	T	t	F
T	t	t	F
t	t	t	F
F	F	F	F

$\vee$	T	t	F
T	t	t	t
t	t	t	t
F	t	t	F

$\Rightarrow$	T	t	F
T	t	t	F
t	t	t	F
F	t	t	t

	$\neg_1$	$\neg_2$
T	F	F
t	F	t
F	T	T

	$\circ_1$	$\circ_2$
T	T	T
t	F	T
F	T	T

The consequence relation in M_0 is defined as follows:

$$\Gamma \vDash_{M_0} \varphi$$

if, for every valuation w over M_0, $w(\varphi) \in D$ whenever $w(\gamma) \in D$ for every $\gamma \in \Gamma$. Observe that a valuation w over M_0 is a mapping $w : L(C) \to \{T, t, F\}$ induced recursively by the operations of the matrix M_0 (technically, w is an homomorphism of C-algebras, see Chapter 3). This kind of logics, called matrix logics (see [280]), will be discussed with more detail in Chapter 9.

The consequence system into which $\mathcal{C}_{\mathbf{Ci}}$ is to be translated is $\langle C, \vDash_{M_0} \rangle$. Next we define the translations. Let Tr be the family of all maps

$$f : L(C^\circ) \to L(C)$$

such that:

- $f(\xi) = \xi$, for $\xi \in \Xi$;
- $f(\varphi \# \psi) = (f(\varphi) \# f(\psi))$, for every $\# \in \{\wedge, \vee, \Rightarrow\}$;
- $f(\neg\varphi) \in \{(\neg_1 f(\varphi)), (\neg_2 f(\varphi))\}$;
- $f(\circ\varphi) \in \{(\circ_1 f(\varphi)), (\circ_2 f(\varphi))\}$;
- if $f(\neg\varphi) = (\neg_1 f(\varphi))$ then $f(\circ\varphi) = (\circ_2 f(\varphi))$;
- if $f(\neg\varphi) = (\neg_2 f(\varphi))$ then $f(\circ\varphi) = (\circ_1 f(\varphi))$.

It can be proved that

$$P = \langle \{\mathcal{C}_f\}_{f \in Tr}, Tr \rangle$$

where $\mathcal{C}_f = \langle C, \vDash_{M_0} \rangle$ for every $f \in Tr$, is a possible–translations semantics for $\mathcal{C}_{\mathbf{Ci}}$ (see Proposition 9.2.20 in Chapter 9).

Note that the semantics of the component consequence systems $\langle C, \vDash_{M_0} \rangle$ of P is truth-functional, in contrast with the non-truth-functional semantics of the initial consequence system $\mathcal{C}_{\mathbf{Ci}}$. That is, a non-truth-functional logic can be characterized by a family of translations into a truth-functional logic. ∇

A more detailed account of the possible-translations technique will be given in Chapter 9. Non-truth-functional logics will be again studied in this book, but under a different perspective, in Chapter 5.

Chapter 2

Splicing logics: Syntactic fibring

As we have seen in Chapter 1, the pursuit for combining logics is dictated by philosophical considerations as well as by practical ones; even if contemporary computer science will gain much on seeing combinations of deducibility relations as solutions to problems in software engineering, security protocols and so on, the purely theoretical interest on combining logics is also very relevant.

As a toy example, let us take the case of someone willing to combine knowledge and obligation in a unique reasoning system. Of course, there are two separated traditions about such distant conceptions that should be harmonized into a bigger system. We depart from the assumption that there is a logic for knowledge (an epistemic logic) as well as a logic for obligation (a deontic logic). Philosophically, they are traditionally attached to completely independent perspectives. In order to combine them, some constraints are in order; for example, we want that the combined new logic will at least guarantee the following requirements:

1. Embedding: the logic machinery of the component systems has to be available in the bigger system as well.

2. Minimality: besides embedding, we do not want any undesirable elements in the combined system; so, for instance, we should be able to talk about the conjoint properties of knowledge and obligation, but nothing more (obviously whether or not the combined system permits to talk about *all* the properties of the combination is another problem which will be dealt with in different guises in this book).

So, for example, in what concerns the embedding requirement, in the combined system we should still be talking about knowledge and obligation *per se*, besides referring to the intricacies of each one.

To what concerns the minimality requirement, to be a bit more concrete, suppose that knowledge is rendered by the operator K in a system S_1 and obligation is represented by the operator O in a system S_2. In the combined system $S = S_1 \cup S_2$, we would be interested in referring to:

- $OK\varphi$ to mean "it is obligatory to know φ";

  as well as

- $KO\varphi$ to mean "it is known that φ is obliged";

but we do not know the relationship between them, that is, we do not know, for example, if $OK\varphi$ implies $KO\varphi$ or the other way around.

The answer to these questions depends upon the minimal coherent relationship that holds between the epistemic and the deontic settings.

The operation of fibring, in a sense, will provide a minimal epistemic deontic logic, but further relations can be added to the resulting combined logic. However, how much and which of such interactions should be introduced in a coherent or meaningful way is a philosophical and not a logical problems. Fibring would be useful as an environment for representing them, but never for deciding which properties should be added.

Taking into account our example above, it is clear that a richer language for talking about knowledge and obligation in an integrated way will be necessary. When defining fibring of logics, it is thus essential to start with a rigorous notion of signature. That is, we should provide the main symbols that allow the construction of formulas. The crucial idea of fibring two logics is that a formula can involve symbols from both logics. In our example, in the fibring we want to have formulas involving both K and O. For more details on this example see Section 11.2 of Chapter 11.

In this chapter we avoid dealing with quantification. This issue will be left for Chapter 6 and Chapter 7. Instead, fibring is here defined in a simple context concentrating our attention on propositional based logics only, but including modal logics, intuitionistic logic and many-valued logics. As the reader will see this is already a rich setting where many relevant concepts can be treated.

The combined system will consist of a more complex deductive system which by its turn will require a more sophisticated semantic interpretation. We have to consider fibring of deductive systems as well as fibring of semantic structures. In this chapter we concentrate on the fibring of deductive systems, leaving the fibring of semantic structures to Chapter 3.

From the deductive point of view, we assume that the logics are endowed with Hilbert calculi. Hence, we adopt what is called an *homogeneous setting* (an heterogeneous setting, in contrast, would allow, for instance, the fibring of a Hilbert calculus with a sequent calculus or tableau systems, as will be discussed in Chapter 4). It is convenient to consider such an homogeneous setting because it is easier to compose systems endowed with Hilbert-like presentations, taking profit that most logics of interest are axiomatized by Hilbert calculi.

In the introductory chapters, in particular in the present one, we mainly adopt a set-theoretic perspective which is easier to begin with. We refer to the categorical approach as side comments to prepare the reader for other chapters (about fibring of higher order logics and modulated fibring) where it is almost compulsory to use categories in order to make definitions clearer and shorter.

In Section 2.1, we introduce the notions of signature, language including schema variables and fibring of signatures. In Section 2.2, we define Hilbert calculi and their fibring. We also show that derivations are preserved by fibring. We illustrate the concepts with several examples including classical logic, modal logics (**K**, **S4** and **B**), intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 2.3, we discuss several preservation results namely: metatheorem of modus ponens, metatheorem of deduction, metatheorem of congruence, careful-reasoning-by-cases and interpolation. In Section 2.4 we present some final remarks.

This chapter capitalizes on [237, 240] for Hilbert calculus, on [282] for the preservation of metatheorems, and on [53] for the preservation of interpolation.

2.1 Language

Defining the language of a logic is to introduce the set of its formulas. Each formula at a certain level of abstraction is just a well-formed finite sequence of symbols. The allowed symbols are the ones that are included in the signature. Propositional signatures are different from first-order and higher-order signatures. We start by indicating the general form of a signature for the propositional based logics. We adopt an algebraic approach to the definition of signature seeing the symbols as operations.

Definition 2.1.1 A *signature* C is a countable family of sets C_k where $k \in \mathbb{N}$. $\triangledown$

The elements of each C_k are called *constructors* or *connectives* of arity k. Constructors of arity 0 are often called *constants* and constructors of arity 1 and 2 are respectively *unary* and *binary* constructors. In general, there is only a finite number of non-empty sets in a signature C and C_k is usually finite for $k > 0$.

In the following examples we present some well-known signatures.

Example 2.1.2 Taking a countable set $\mathbb{P} = \{p_n : n \in \mathbb{N}\}$ (of propositional symbols), we can consider the following signatures:

- Classical signature with propositional symbols:

  $C_0 = \mathbb{P}$, $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow\}$, $C_k = \emptyset$ for $k > 2$;

- Intuitionistic signature with propositional symbols:

  $C_0 = \mathbb{P}$, $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow, \wedge, \vee\}$, $C_k = \emptyset$ for $k > 2$;

- Modal signature with propositional symbols:

  $C_0 = \mathbb{P}$, $C_1 = \{\neg, \Box\}$, $C_2 = \{\Rightarrow\}$, $C_k = \emptyset$ for $k > 2$. ▽

Observe that the usual description of propositional logics given in the textbooks just includes $\mathbb{P}$ in the signature, whereas the connectives are considered as being reserved symbols. We adopt here a different perspective, where symbols are better described as operations. This approach is more convenient for combining logics, where we must be able to manipulate carefully all the symbols. Note also that for the same logic we can consider several signatures. For instance, for propositional classical logic we obtain a different signature for each adequate set of connectives.

It should be clear that the present approach to signatures is equivalent to consider just one set of connectives together with a map indicating the arity of each symbol. However, the grouping of symbols by arity is more useful for fibring.

We also stress that at this point we are not giving any properties for the connectives. They will be imposed later on both at the deductive and the semantic levels. For the moment connectives are just symbols. Hence, we may have two completely different logics with the same signatures.

In classical logic we only consider negation $\neg$ and implication $\Rightarrow$ since the other connectives can be seen as abbreviations. That is not the case for intuitionistic logic (see [232]) and so we also have conjunction $\wedge$ and disjunction $\vee$ as primitive symbols. Observe that at the signature level we do not distinguish between the classical and the intuitionistic implications. In modal logic we only consider the modality $\Box$. The modality $\Diamond$ can also be seen as an abbreviation.

Since our main purpose is to combine logics, we need to introduce some additional elements to the propositional languages to be considered herein. Thus, from now on we will consider a fixed denumerable set

$$\Xi = \{\xi_n : n \in \mathbb{N}^+\}$$

of symbols called *schema variables*, where $\mathbb{N}^+$ denotes the set of positive integers. The schema variables can be freely substituted by formulas of the given logics. The inclusion of such variables allow the introduction of schematic rules in the fibring setting.

For most purposes the set Ξ can be assumed to be fixed. The exception is when dealing with interpolation, in Section 2.3 and modulated fibring, in Chapter 8.

Remark 2.1.3 The difference between propositional symbols and schema variables will be clear below when investigating the combination of logics. In this context, it is more convenient to keep the propositional symbols of the logics as constants, separated from the schema variables, as it was done in Example 2.1.2.

Anyway, the inclusion of the propositional symbols as constants can be useful even in purely propositional contexts. In fact, some situations could require the combination of two propositional logics without sharing the propositional symbols.

Our approach is different from some approaches in logic where axioms and inference rules are written without schema variables. Consider the classical propositional logic **CPL** defined in a Hilbert-style. For instance, the Hilbert calculus usually includes the axioms

$$(\varphi_1 \Rightarrow (\varphi_2 \Rightarrow \varphi_1))$$

for every formulas φ_1 and φ_2 belonging to the language of **CPL**. Assume that we want to combine **CPL** with another logic. Then the axiom could not be used with formulas of the other logic.

Our approach is also different from the ones using propositional variables. Considering again the classical propositional logic **CPL**, we can give the axiom

$$(p_1 \Rightarrow (p_2 \Rightarrow p_1))$$

where p_1 and p_2 are propositional symbols in $\mathbb{P}$. Moreover, a rule of uniform substitution is also included saying that uniformly substituting propositional variables by formulas in a theorem will also give a theorem. The main drawback of this approach is when we consider derivation from hypotheses [24]. We will come back to this topic later on. $\triangledown$

Example 2.1.4 It is possible to recast Example 2.1.2 by omitting in the signatures the set $\mathbb{P}$ of propositional symbols. Thus, we can consider the following signatures:

- Classical signature: $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow\}$, and $C_k = \emptyset$ in any other case;
- Intuitionistic signature: $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow, \wedge, \vee\}$, and $C_k = \emptyset$ in any other case;
- Modal signature: $C_1 = \{\neg, \Box\}$, $C_2 = \{\Rightarrow\}$, and $C_k = \emptyset$ in any other case. $\triangledown$

We are now ready to define the language over a given signature.

Definition 2.1.5 Let C be a signature, and assume the fixed set Ξ of schema variables. The *language* over C is the set $L(C)$ inductively defined as follows:

- $\xi \in L(C)$ for every $\xi \in \Xi$;
- $c \in L(C)$ for every $c \in C_0$;
- $(c(\varphi_1, \ldots, \varphi_k)) \in L(C)$ whenever $c \in C_k$, $k \geq 1$ and $\varphi_1, \ldots, \varphi_k \in L(C)$. $\triangledown$

The symbols "(", ")" and "," in expressions like $(c(\varphi_1, \ldots, \varphi_k))$ are auxiliary symbols. The elements of $L(C)$ are called *formulas*. A *ground formula* is a formula that does not involve schema variables and $gL(C)$ denotes the set of all ground formulas in $L(C)$. Technically, $L(C)$ is the free algebra over C generated by Ξ (to be defined in Section 3.1 of Chapter 3).

Recall Example 2.1.2. An example of formula in the language over the classical signature with propositional symbols is

$$(\neg(\Rightarrow(\xi_1, p)))$$

where $p \in C_0$ and $\xi_1 \in \Xi$. For convenience we adopt the more usual infix notation for connectives. This formula then becomes

$$(\neg(\xi_1 \Rightarrow p)).$$

In classical logic we can consider the usual abbreviations

$$(\varphi_1 \wedge \varphi_2) =_{\text{def}} (\neg(\varphi_1 \Rightarrow (\neg\varphi_2)))$$

$$(\varphi_1 \vee \varphi_2) =_{\text{def}} (((\neg\varphi_1) \Rightarrow \varphi_2))$$

$$(\varphi_1 \Leftrightarrow \varphi_2) =_{\text{def}} ((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)).$$

In modal logic, we can consider the abbreviation

$$(\Diamond\varphi) =_{\text{def}} (\neg(\Box(\neg\,\varphi))).$$

The formulas $(\Box\varphi)$ and $(\Diamond\varphi)$ can be read as necessarily φ and possibly φ, respectively.

The signatures introduced in Example 2.1.4 are useful in a purely propositional context for combining logics, and they are simpler than the corresponding signatures described in Example 2.1.2. As mentioned above, the schema variables play the role of the atomic formulas in the usual presentations of propositional languages. The effects, at the semantical level, of the simpler case which avoids the use of the set $\mathbb{P}$ of propositional symbols, will be analyzed in Chapter 3 for some concrete examples.

From now on, and for the sake of simplicity, most of the concrete examples will be written in signatures not including the set $\mathbb{P}$ of propositional symbols, using the signatures described in Example 2.1.4. Of course, all the examples below can be recasted by including the set $\mathbb{P}$ of propositional symbols as constants.

The inductive nature of $L(C)$ is very useful because several properties over $L(C)$ can be proved by induction. Moreover, although no restrictions were imposed on the cardinality of the set of connectives for each arity, in each formula there is always a finite number of connectives. The same applies when we are dealing with a finite set of formulas.

An essential ingredient is to be able to produce new formulas by replacing the schema variables by formulas. For this purpose we need the concept of substitution. Substitutions are used when dealing with structural calculi.

Definition 2.1.6 A *substitution* over a signature C is a map $\sigma : \Xi \to L(C)$. Every substitution σ can be extended to a unique mapping $\widehat{\sigma} : L(C) \to L(C)$ such that

- $\widehat{\sigma}(c) = c$, if $c \in C_0$;
- $\widehat{\sigma}(c(\psi_1, \ldots, \psi_k)) = (c(\widehat{\sigma}(\psi_1), \ldots, \widehat{\sigma}(\psi_k)))$, if $c \in C_k$, $\psi_1, \ldots, \psi_k \in L(C)$ and $k > 0$. ▽

An example of a substitution is the following map: $\sigma(\xi_n) = \xi_{n+2}$ which is just a renaming of the schema variables. The *instance* of a formula φ by a substitution σ is denoted by $\widehat{\sigma}(\varphi)$. Sometimes, and when there is no risk of confusion, we may identify a substitution σ with its extension $\widehat{\sigma}$. If Γ is a set of formulas then $\sigma(\Gamma)$ denotes the set $\{\sigma(\gamma) : \gamma \in \Gamma\}$. We say that a substitution σ is *ground* when $\sigma(\xi_n)$ is a ground formula for every $n \in \mathbb{N}^+$.

As a first step to defining fibring of logics, we must define the fibring of signatures. The following notation will be useful from now on. We denote by $C' \cap C''$ the signature such that $(C' \cap C'')_k = C'_k \cap C''_k$ for every $k \in \mathbb{N}$. We write $C' \cap C'' = \emptyset$ whenever $(C' \cap C'')_k = \emptyset$ for every k.

Definition 2.1.7 The *fibring of signatures* C' and C'' is the signature

$$C' \cup C''$$

such that $(C' \cup C'')_k = C'_k \cup C''_k$ for every $k \in \mathbb{N}$. ▽

The fibring is said to be *unconstrained* when $C' \cap C'' = \emptyset$. Otherwise, the fibring is said to be *constrained*. Note that, in general, $L(C' \cup C'') \neq L(C') \cup L(C'')$.

Signatures can be compared. We say that

$$C \leq C'$$

when $C_k \subseteq C'_k$ for every $k \in \mathbb{N}$. Of course $C' \leq C' \cup C''$, $C'' \leq C' \cup C''$ and $C' \cap C'' \leq C'$, $C' \cap C'' \leq C''$.

Example 2.1.8 Let C' and C'' be two modal signatures as in Example 2.1.4, with different modalities denoted respectively by $\Box'$ and $\Box''$, and sharing the propositional symbols $\neg$ and $\Rightarrow$. Then the fibring of C' and C'' is the signature $C' \cup C''$ where:

- $(C' \cup C'')_1 = \{\neg, \Box', \Box''\}$;
- $(C' \cup C'')_2 = \{\Rightarrow\}$;
- $(C' \cup C'')_k = \emptyset$ in any other case. ▽

Example 2.1.9 Consider again the Example 2.1.8, but now assume that

$$C'_0 = \mathbb{P} = C''_0$$

(recall Example 2.1.2). Then the fibring of C' and C'' while sharing the propositional symbols $\mathbb{P}$, $\neg$ and $\Rightarrow$ is the signature $C' \cup C''$ where:

- $(C' \cup C'')_0 = \mathbb{P}$;
- $(C' \cup C'')_1 = \{\neg, \Box', \Box''\}$;
- $(C' \cup C'')_2 = \{\Rightarrow\}$;
- $(C' \cup C'')_k = \emptyset$ for $k > 2$.

Assuming $p_1, p_2 \in \mathbb{P}$, an example of a ground formula in the language $L(C' \cup C'')$ is the following:

$$(\Box'((\Box''(p_1 \Rightarrow p_2)) \Rightarrow ((\Box'' p_1) \Rightarrow (\Box'' p_2)))).$$

Note that this formula is not an element of $L(C') \cup L(C'')$. ∇

Remark 2.1.10 Fibring can be presented in a categorial setting: the notion of fibring corresponds to an universal construction in an appropriated category. Establishing the category of signatures is the first step in this approach.

A signature morphism

$$h : C \to C'$$

is a family of maps $h_k : C_k \to C'_k$ where $k \in \mathbb{N}$. Signatures and their morphisms constitute the category **Sig**, with identity and composition of maps defined on each arity. **Sig** is (small) cocomplete, that is, it is closed under (small) coproducts and pushouts.

The fibring $C' \cup C''$ of C' and C'' is a pushout of the inclusion morphisms $h' : C' \cap C'' \to C'$ and $h'' : C' \cap C'' \to C''$. Figure 2.1 describes this situation.

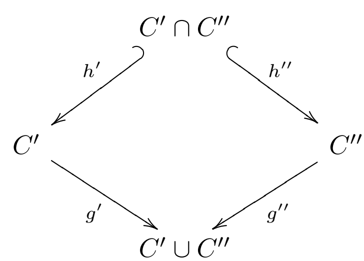


Figure 2.1: Fibring of signatures as a pushout in **Sig**

Recall from Chapter 1 that this pushout can be obtained in two steps. First we consider the coprodut $C' \oplus C''$ of signatures C' and C'' and then the coequalizer q of $i' \circ h'$ and $i'' \circ h''$ where $g' = q \circ i'$ and $g'' = q \circ i''$ (see Figure 2.2).

The particular case of unconstrained fibring, when $C' \cap C'' = \emptyset$, just corresponds to the coproduct of C' and C''. ∇

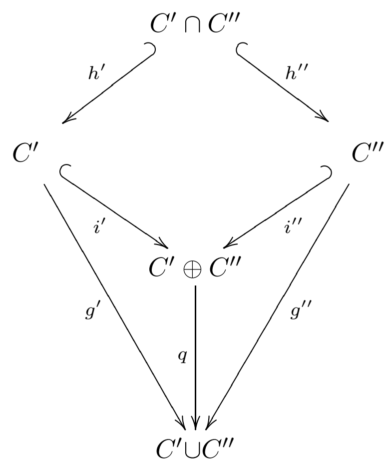


Figure 2.2: Construction of a pushout in **Sig**

2.2 Hilbert calculi

A calculus uses only symbolic manipulation of formulas to establish which are the consequences of a set of formulas. Preferably, an evidence that a formula is a consequence of set of formulas should be provided in a finite number of steps. The main concept in a calculus is the one of derivation. There are different ways to present the notion of derivation. Herein, we concentrate our attention on Hilbert calculi. The name of these calculi comes from the German mathematician David Hilbert who defended the axiomatic approach in Mathematics [148]. Following the axiomatic approach, every mathematical theory should be presented by axioms and rules. Although derivations are in general difficult to obtain using Hilbert calculi, they are very easy to describe at a theoretical level.

Definition 2.2.1 An *inference rule* over a signature C is a pair $r = \langle \Delta, \varphi \rangle$ where $\Delta \cup \{\varphi\} \subseteq L(C)$. $\triangledown$

An inference rule is called *axiom* if $\Delta = \emptyset$ and *rule* if $\Delta \neq \emptyset$. A rule where Δ is a finite set is called *finitary*. We do not consider non-finitary rules so, in the sequel, when we refer to a rule $\langle \Delta, \varphi \rangle$ we always assume that Δ is a finite set. The elements of Δ are called *premises* and φ is called *conclusion*.

The schema variables are essential for inference rules. Whenever a schema variable ξ occurs in an inference rule, it is possible to uniformly substitute ξ by any formula. Thus, a single rule encompasses infinite instances.

We are now ready to define in a very abstract way the notion of Hilbert calculus.

Definition 2.2.2 A *Hilbert calculus* is a pair

$$H = \langle C, R \rangle$$

where C is a signature and R is a set of inference rules over C. ▽

Example 2.2.3 Recall the classical signature introduced in Example 2.1.4. A Hilbert calculus for classical logic includes besides that signature, the following inference rules:

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, (((\neg \xi_1) \Rightarrow (\neg \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$.

Observe that there are three axioms and a rule (usually called modus ponens). Each axiom provides a pattern. The instances of the first axiom have two implications such that the antecedent formula (instance of ξ_1) of the first implication is the consequent formula of the second implication. We denote the i-th axiom, $i = 1, 2, 3$, by Ax_i and the rule by MP . ▽

Another useful illustration can be given for modal logic. For details on modal logics see, for instance, [24].

Example 2.2.4 Recall the modal signature introduced in Example 2.1.4. A Hilbert calculus for propositional normal modal logic **K** includes, besides that signature, all the rules for classical logic (see Example 2.2.3) plus the following inference rules:

- $\langle \emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2))) \rangle$;
- $\langle \{\xi_1\}, (\Box\xi_1) \rangle$.

Since the Hilbert calculus for modal logic **K** is an extension of the Hilbert calculus for classical logic, propositional reasoning is guaranteed. The new axiom, called *normalization* or K characterizes normal modal logic. The new rule is called *necessitation* or Nec . ▽

Several normal modal logics have been analyzed. The Hilbert calculi for such logics include the inference rules for **K** plus one or more specific axioms.

Example 2.2.5 The Hilbert calculus for modal logic **S4** (semantically corresponding to reflexive and transitive frames) has the same inference rules as modal logic **K** plus two axioms:

- $\langle\emptyset, ((\Box\xi_1) \Rightarrow \xi_1)\rangle$;
- $\langle\emptyset, ((\Box\xi_1) \Rightarrow (\Box(\Box\xi_1)))\rangle$.

The first axiom is known as the T axiom (for reflexivity) and the second one is called the 4 axiom (for transitivity). Another well known normal modal logic is **B**. The Hilbert calculus for **B** (semantically corresponding to symmetric frames) includes besides the inference rules for **K** also the axiom:

- $\langle\emptyset, (\xi_1 \Rightarrow (\Box(\neg(\Box(\neg\,\xi_1)))))\rangle$.

This axiom is called the B axiom. Using the abbreviation introduced after Definition 2.1.5, it can also be written as $(\xi_1 \Rightarrow (\Box(\Diamond\xi_1)))$. ∇

Another interesting example is the case of intuitionistic logic which is the essence of the so called constructivism. For details on intuitionistic logic see [232].

Example 2.2.6 Recall the intuitionistic signature introduced in Example 2.1.4. A Hilbert calculus for intuitionistic logic includes the following inference rules:

- $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1))\rangle$;
- $\langle\emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3)))\rangle$;
- $\langle\emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_1)\rangle$;
- $\langle\emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_2)\rangle$;
- $\langle\emptyset, ((\xi_3 \Rightarrow \xi_1) \Rightarrow ((\xi_3 \Rightarrow \xi_2) \Rightarrow (\xi_3 \Rightarrow (\xi_1 \wedge \xi_2))))\rangle$;
- $\langle\emptyset, (\xi_1 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle\emptyset, (\xi_2 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle\emptyset, ((\xi_1 \Rightarrow \xi_3) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow ((\xi_1 \vee \xi_2) \Rightarrow \xi_3)))\rangle$;
- $\langle\emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_1 \Rightarrow (\neg\xi_2)) \Rightarrow (\neg\xi_1)))\rangle$;
- $\langle\emptyset, (\xi_1 \Rightarrow ((\neg\xi_1) \Rightarrow \xi_2))\rangle$;
- $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$.

The Hilbert calculus for intuitionistic logic is not an extension of the one for classical logic. The implication does not have all the properties of the classical implication. The explicit axioms for conjunction and disjunction will be justified later on. It is worthwhile to point out at this stage that we cannot obtain conjunction and disjunction as abbreviations using implication and negation. Also to be understood later on is the fact that the axioms for conjunction and disjunction impose that they are respectively an infimum and a supremum in an adequate type of structure. ∇

Also of interest are the so called many-valued logics. They provide nice examples for understanding fibring. For details on many-valued logics see, for instance, [140, 137].

Example 2.2.7 The Gödel signature is the same as the intuitionistic one, and so we can consider either the intuitionistic signature of Example 2.1.2 or that of Example 2.1.4. A Hilbert calculus for 3-valued Gödel logic includes the intuitionistic inference rules plus the axiom:

- $\langle \emptyset, (((\neg\, \xi_1) \Rightarrow \xi_2) \Rightarrow (((\xi_2 \Rightarrow \xi_1) \Rightarrow \xi_2) \Rightarrow \xi_2)) \rangle$. $\triangledown$

Example 2.2.8 The Łukasiewicz signature is the same as the classical one. Thus, we can choose the classical signature defined in Example 2.1.2 or the one introduced in Example 2.1.4. A Hilbert calculus for 3-valued Łukasiewicz logic, Ł$_3$, includes, besides the signature, the following inference rules:

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, ((\neg\, \xi_1) \Rightarrow (\neg\, \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, (((\xi_1 \Rightarrow (\neg\, \xi_1)) \Rightarrow \xi_1) \Rightarrow \xi_1) \rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$.

Note that the second axiom is not the axiom Ax_2 of the Hilbert calculus for classical logic presented above.

It is well-known that there exists a hierarchy of n-valued Łukasiewicz logics, Ł$_n$, for each $n \geq 3$. $\triangledown$

For more details on Gödel logic and Łukasiewicz logic see [232] (named after Kurt Gödel and Jan Łukasiewicz, respectively).

The next example presents a Hilbert calculus for paraconsistent logic $\mathfrak{C}_1$, introduced by da Costa in [73]. This logic has the peculiarity that it does not admit a truth-functional semantics. In particular, the semantic structures to be given in Chapter 3 are not adequate for it. In Chapter 5, we will introduce suitable semantic structure for non-truth-functional logics.

Example 2.2.9 The signature for the paraconsistent logic $\mathfrak{C}_1$ includes, besides the signature of intuitionistic logic, the constants **t** and **f**. A Hilbert calculus for $\mathfrak{C}_1$ is as follows:

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_1) \rangle$;

- $\langle\emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_2)\rangle$;
- $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow (\xi_1 \wedge \xi_2)))\rangle$;
- $\langle\emptyset, (\xi_1 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle\emptyset, (\xi_2 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle\emptyset, ((\xi_1 \Rightarrow \xi_3) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow ((\xi_1 \vee \xi_2) \Rightarrow \xi_3)))\rangle$;
- $\langle\emptyset, ((\neg(\neg\, \xi_1)) \Rightarrow \xi_1)\rangle$;
- $\langle\emptyset, (\xi_1 \vee (\neg\, \xi_1))\rangle$;
- $\langle\emptyset, (\xi_1^\circ \Rightarrow (\xi_1 \Rightarrow ((\neg\, \xi_1) \Rightarrow \xi_2)))\rangle$;
- $\langle\emptyset, ((\xi_1^\circ \wedge \xi_2^\circ) \Rightarrow (\xi_1 \wedge \xi_2)^\circ)\rangle$;
- $\langle\emptyset, ((\xi_1^\circ \wedge \xi_2^\circ) \Rightarrow (\xi_1 \vee \xi_2)^\circ)\rangle$;
- $\langle\emptyset, ((\xi_1^\circ \wedge \xi_2^\circ) \Rightarrow (\xi_1 \Rightarrow \xi_2)^\circ)\rangle$;
- $\langle\emptyset, (\mathbf{t} \Leftrightarrow (\xi_1 \Rightarrow \xi_1))\rangle$;
- $\langle\emptyset, (\mathbf{f} \Leftrightarrow (\xi_1^\circ \wedge (\xi_1 \wedge (\neg\, \xi_1))))\rangle$;
- $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$;

In the rules above φ° is an abbreviation for $(\neg(\varphi \wedge (\neg\, \varphi)))$. ∇

Given a Hilbert calculus, formulas can be deduced from sets of formulas (the hypotheses). The way to do so is through derivation. Derivations should be constructive in the sense that they explain step by step, in finite time, how a formula can be deduced from a set of hypotheses.

Definition 2.2.10 A formula φ is *derivable* from a set of formulas Γ in a Hilbert calculus H if there is a finite sequence

$$\varphi_1 \cdots \varphi_n$$

of formulas such that:

- φ_n is φ;
- for $i = 1, \ldots, n$, each φ_i is either an element of Γ, or there exists a substitution σ and an inference rule $\langle\Delta, \psi\rangle$ in H such that $\sigma(\Delta) \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$ and φ_i is $\sigma(\psi)$.

The sequence $\varphi_1 \dots \varphi_n$ is a *derivation* of φ from Γ in H. We write

$$\Gamma \vdash_H \varphi$$

to denote that φ is derivable from Γ in H . ∇

We denote by $\Gamma^{\vdash_H}$ the set of formulas derivable from (the set of hypotheses) Γ in H. When presenting derivations we usually add a justification for each formula. We use Hyp as the justification for an hypothesis and we use the name of the inference rule when the formula is an instance of the conclusion of the inference rule. If the inference rule is not an axiom we also add the positions of the formulas in the sequence that are instances of the premises. Observe the finitary character of the derivation in the sense that with a finite number of steps we are able to show that φ is derivable from Γ.

Note that we do not allow substitutions on hypotheses (the elements of Γ). Indeed, such substitutions do not make sense. For instance, from $\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}$ we want to be able to prove ξ_2, but not every formula as it would be possible by substitution on ξ_1. When $\emptyset \vdash_H \varphi$ we say that φ is a *theorem* and just write $\vdash_H \varphi$. As usual, $\Gamma, \psi \vdash_H \varphi$ and $\psi \vdash_H \varphi$ will stand, for $\Gamma \cup \{\psi\} \vdash_H \varphi$ and $\{\psi\} \vdash_H \varphi$ respectively.

A Hilbert calculus $H = \langle C, R\rangle$ induces the closure operator $\vdash_H$ and the consequence system $\mathcal{C}(H) = \langle C, \vdash_H\rangle$. We write $\Gamma^{\vdash_H}$ instead of $\vdash_H (\Gamma)$.

Proposition 2.2.11 *A Hilbert calculus $H = \langle C, R\rangle$ induces a compact and structural consequence system $\mathcal{C}(H) = \langle C, \vdash_H\rangle$.*

Proof. Let $\Gamma, \Gamma_1, \Gamma_2 \subseteq L(C)$.

Extensiveness: $\Gamma \vdash_H \varphi$ for every $\varphi \in \Gamma$ thus $\Gamma \subseteq \Gamma^{\vdash_H}$.

Monotonicity: Let $\Gamma_1 \subseteq \Gamma_2$. We prove by induction on the length of a derivation of φ from Γ_1 in H that $\Gamma_2 \vdash_H \varphi$.

Base: When φ has a derivation from Γ_1 with length 1, then $\varphi \in \Gamma_1$ (thus $\varphi \in \Gamma_2$) or $\varphi = \sigma(\psi)$ for some axiom $\langle \emptyset, \psi\rangle \in R$ and substitution σ. In both cases $\Gamma_2 \vdash_H \varphi$.

Step: Assume φ has a derivation of length $n + 1$ from Γ_1. The interesting case corresponds to $\varphi = \sigma(\psi)$ for some rule $\langle \Delta, \psi\rangle \in R$ and substitution σ such that every formula of $\sigma(\Delta)$ occurs previously in the derivation. By the induction hypothesis, every formula in $\sigma(\Delta)$ has a derivation from Γ_2. From them and rule $\langle \Delta, \psi\rangle$ we get a derivation of φ from Γ_2, thus $\Gamma_2 \vdash_H \varphi$.

Idempotence: Induction in the length of derivations to prove that if $\Gamma^{\vdash_H} \vdash_H \varphi$ then $\Gamma \vdash_H \varphi$, for every $\varphi \in L(C)$.

Base: When φ has a derivation from $\Gamma^{\vdash_H}$ with length 1, then $\varphi \in \Gamma^{\vdash_H}$ or $\varphi = \sigma(\psi)$ for some axiom $\langle \emptyset, \psi\rangle \in R$ and substitution σ. In both cases $\Gamma \vdash_H \varphi$.

Step: When φ has a derivation of length $n + 1$ from $\Gamma^{\vdash_H}$, the proof is similar to the one presented on the induction proof above.

Compactness follows from the fact that derivations are finite sequences.

Structurality: Induction in the length of derivations to prove that for every $\varphi \in L(C)$ and substitution σ, if $\Gamma \vdash_H \varphi$ then $\sigma(\Gamma) \vdash_H \sigma(\varphi)$.

Base: When φ has a derivation from Γ with length 1, then $\varphi \in \Gamma$ or $\varphi = \sigma'(\psi)$ for some axiom $\langle \emptyset, \psi \rangle \in R$ and substitution σ'. In the first case, $\sigma(\varphi) \in \sigma(\Gamma)$, hence $\sigma(\Gamma) \vdash_H \sigma(\varphi)$. In the second case, consider the substitution $\sigma'' : \Xi \to L(C)$ such that $\sigma''(\xi) = \sigma(\sigma'(\xi))$. It is easy to prove that $\sigma''(\psi) = \sigma(\sigma'(\psi))$. Hence, $\sigma''(\psi) = \sigma(\varphi)$ and therefore $\sigma(\Gamma) \vdash_H \sigma(\varphi)$.

Step: Assume that φ has a derivation of length $n+1$ from Γ. The interesting case corresponds to $\varphi = \sigma(\psi)$ for some rule $\langle \Delta, \psi \rangle \in R$ and some substitution σ such that the formulas in $\sigma(\Delta)$ previously occur in the derivation. Using the induction hypothesis and the substitution σ'' as above, we conclude that $\sigma(\Gamma) \vdash_H \sigma(\varphi)$. ◁

In general, $\vdash_H$ is not a topological (Kuratowski) closure operator. In fact, $\emptyset^{\vdash_H} \neq \emptyset$ whenever the calculus H includes axioms. Moreover, it is often the case that $(\Gamma_1 \cup \Gamma_2)^{\vdash_H} \neq \Gamma_1^{\vdash_H} \cup \Gamma_2^{\vdash_H}$.

Observe that the converse of Proposition 2.2.11 is also true. Thus Hilbert calculi and compact and structural consequence systems are essentially the same. Of course, this is not the original idea of Hilbert calculi. This motivates the following definition.

Definition 2.2.12 A Hilbert calculus $H = \langle C, R \rangle$ is said to be *recursive* if $L(C)$ is a recursive set and, for each recursive set $\Gamma \subseteq L(C)$, the set $\Gamma^{\vdash_H}$ is recursively enumerable. ▽

Note that all the examples presented above correspond to recursive Hilbert calculi. The importance of such Hilbert calculi is well known since the main objective is to work with logics that are at least semi-decidable. Proving that a Hilbert calculus is recursive can be done using the projection lemma of computability once we have shown that unary relations corresponding to axioms are recursive and that (k+1)-ary relations corresponding to k-ary inference rules are also recursive. Moreover, a preliminary step is to make the set of formulas a Gödel domain so that we can encode formulas into natural numbers.

We are now ready to investigate the notion of fibring. Fibring of Hilbert calculi is easy to introduce since our inference rules are schematic.

Definition 2.2.13 The *fibring of Hilbert calculi* $H' = \langle C', R' \rangle$ and $H'' = \langle C'', R'' \rangle$ is the Hilbert calculus

$$H' \cup H'' = \langle C, R \rangle$$

where $C = C' \cup C''$ and $R = R' \cup R''$. ▽

The fibring of H' and H'' is *unconstrained* when the fibring of their signatures is unconstrained, that is, $C' \cap C'' = \emptyset$, and it is *constrained* otherwise. It is worth noting that the fibring of recursive Hilbert calculi is also recursive.

Example 2.2.14 Recall Example 2.2.5. Let $H_{\mathbf{S4}}$ and $H_{\mathbf{B}}$ be two Hilbert calculi for modal logics **S4** and **B** respectively, sharing the same signature (just one box). The following is a derivation in the fibring $H_{\mathbf{S4}} \cup H_{\mathbf{B}}$ of $H_{\mathbf{S4}}$ and $H_{\mathbf{B}}$.

1	$((\Box\xi_1) \Rightarrow \xi_1)$	T
2	$(((\Box\xi_1) \Rightarrow \xi_1) \Rightarrow (\Box(\neg(\Box(\neg((\Box\xi_1) \Rightarrow \xi_1))))))$	B
3	$(\Box(\neg(\Box(\neg((\Box\xi_1) \Rightarrow \xi_1)))))$	MP1, 2

Hence, the formula $(\Box(\neg(\Box(\neg((\Box\xi_1)\Rightarrow\xi_1)))))$ is a theorem of $H_{\mathbf{S4}} \cup H_{\mathbf{B}}$. Note that the new calculus $H_{\mathbf{S4}} \cup H_{\mathbf{B}}$ is just the Hilbert calculus for a normal modal logic with axioms T, 4 and B, usually known as **S5**. ▽

Example 2.2.15 Recall the previous example and assume that we only want to share the propositional symbols $\neg$ and $\Rightarrow$. Hence, we consider the Hilbert calculus $H'_{\mathbf{S4}}$ defined over the signature C' of Example 2.1.8 and the Hilbert calculus $H''_{\mathbf{B}}$ defined over the signature C'' of the same example. This means that the inference rules explicitly involving modalities must be renamed accordingly. The following is a derivation in the fibring $H'_{\mathbf{S4}} \cup H''_{\mathbf{B}}$ of $H'_{\mathbf{S4}}$ and $H''_{\mathbf{B}}$.

1	$((\Box'\xi_1) \Rightarrow \xi_1)$	T′
2	$(((\Box'\xi_1) \Rightarrow \xi_1) \Rightarrow (\Box''(\neg(\Box''(\neg((\Box'\xi_1) \Rightarrow \xi_1))))))$	B″
3	$(\Box''(\neg(\Box''(\neg((\Box'\xi_1) \Rightarrow \xi_1)))))$	MP1, 2

Thus, $(\Box''(\neg(\Box''(\neg((\Box'\xi_1) \Rightarrow \xi_1)))))$ is a theorem of $H'_{\mathbf{S4}} \cup H''_{\mathbf{B}}$. This formula includes connectives from both calculi. Note that $(\Box'(\neg(\Box'(\neg((\Box''\xi_1)\Rightarrow\xi_1)))))$ is not a theorem of $H'_{\mathbf{S4}} \cup H''_{\mathbf{B}}$. The calculus $H'_{\mathbf{S4}} \cup H''_{\mathbf{B}}$ is a bimodal Hilbert calculus. ▽

Example 2.2.16 Let H be the Hilbert calculus for classical logic presented in Example 2.2.3. The fibring $H \cup H$ is just the Hilbert calculus H. ▽

Example 2.2.17 The unconstrained fibring of the propositional deductive system H' and the Gödel G3 deductive system H'' is the deductive system H such that:

- $C_0 = \{\mathbf{t}', \mathbf{f}'\}$, $C_1 = \{\neg', \neg''\}$, $C_2 = \{\Rightarrow', \Rightarrow'', \wedge'', \vee''\}$, $C_k = \emptyset$ for $k \geq 3$:
- the set R includes all rules for the connectives of both deductive systems.

For instance, two versions $\langle\{\xi_1, (\xi_1 \Rightarrow' \xi_2)\}, \xi_2\rangle$ and $\langle\{\xi_1, (\xi_1 \Rightarrow'' \xi_2)\}, \xi_2\rangle$ of the modus ponens for the propositional and the Gödel implications are included in R. In this case, if we share negation then the fibring collapses to the propositional deductive system. The collapsing problems of fibring will be analyzed in Chapter 8. ▽

Note the importance of having a global set of schema variables. In this way they can be replaced by any formula even one with connectives from both logics.

Of course that would not be the case if the set of schema variables was local to each calculus.

Observe that fibring of Hilbert calculi does introduce interactions between connectives that are not shared. Intended interactions have to be explicitly added to the fibring calculus. As an illustration consider again Example 2.2.15. Assume that we want to impose that $(\Box'\xi)$ always implies $(\Box''\xi)$. Then, the axiom

$$\langle \emptyset, ((\Box'\xi) \Rightarrow (\Box''\xi))\rangle$$

should be added to the Hilbert calculus $H'_{\mathbf{S4}} \cup H''_{\mathbf{B}}$. For another example see Chapter 5

Fibring preserves derivations, that is, we are able to derive in the fibring everything that we could in the original Hilbert calculi. We start by introducing a weakness relation between calculi. Recall Definition 1.1.6.

Definition 2.2.18 The Hilbert calculus $H = \langle C, R\rangle$ is *weaker* than Hilbert calculus $H' = \langle C', R'\rangle$, written $H \leq H'$, if $C \leq C'$ and $\mathcal{C}(H) \leq \mathcal{C}(H')$. ∇

Note that the weakness relation is a partial order, that is, a reflexive, transitive and anti-symmetric relation.

The following result states that the original Hilbert calculi are weaker than their fibring. That is, everything that we derive with the original calculi also can be derived in the fibring.

Proposition 2.2.19 *For every Hilbert calculi H' and H'', the following relationships hold: $H' \leq H' \cup H''$ and $H'' \leq H' \cup H''$.*

Proof. Let $H' = \langle C', R'\rangle$ and $H'' = \langle C'', R''\rangle$. Clearly, $C' \leq C' \cup C''$ and $C'' \leq C' \cup C''$. Moreover, using induction, we easily prove that every derivation of $\varphi' \in L(C')$ from $\Gamma' \subseteq L(C')$ in H' is also a derivation of φ' from Γ' in $H' \cup H''$. Therefore $\Gamma'^{\vdash_{H'}} \subseteq \Gamma'^{\vdash_{H' \cup H''}}$. Similarly, $\Gamma''^{\vdash_{H''}} \subseteq \Gamma''^{\vdash_{H' \cup H''}}$, for every $\Gamma'' \subseteq L(C'')$. $\triangleleft$

Now we will see that the fibring of Hilbert calculi H' and H'' is minimal in the class of all Hilbert calculi that are stronger than H' and H''.

Proposition 2.2.20 *For every Hilbert calculi H, H' and H'', if $H' \leq H$ and $H'' \leq H$ then $H' \cup H'' \leq H$.*

Proof. Let $H = \langle C, R\rangle$, $H' = \langle C', R'\rangle$ and $H'' = \langle C'', R''\rangle$. We have to prove that $C' \cup C'' \leq C$ and $\Gamma^{\vdash_{H' \cup H''}} \subseteq \Gamma^{\vdash_H}$ for every $\Gamma \subseteq L(C' \cup C'')$.
Given that $C' \leq C$ and $C'' \leq C$ it follows that $C' \cup C'' \leq C$.
Since $\Gamma^{\vdash_{H'}} \subseteq \Gamma^{\vdash_H}$ for every $\Gamma \subseteq L(C')$ and $\Delta \vdash_{H'} \psi$ for each $\langle \Delta, \psi\rangle \in R'$, we conclude that $\Delta \vdash_H \psi$ for each $\langle \Delta, \psi\rangle \in R'$. Moreover, by Proposition 2.2.11, $\langle C, {\vdash_H}\rangle$ is a structural consequence system. It follows then that $\sigma(\Delta) \vdash_H \sigma(\psi)$ for each $\langle \Delta, \psi\rangle \in R'$ and substitution σ on C. Similarly, we conclude $\sigma(\Delta) \vdash_H \sigma(\psi)$ for each $\langle \Delta, \psi\rangle \in R''$ and substitution σ on C.

We prove by induction on the length of a derivation of φ from Γ in $H' \cup H''$ that $\Gamma \vdash_H \varphi$, for every $\varphi \in L(C' \cup C'')$ and $\Gamma \subseteq L(C' \cup C'')$.

Base: When φ has a derivation of length 1 from Γ in $H' \cup H''$ then $\varphi \in \Gamma$ or $\varphi = \sigma(\psi)$ for some axiom $\langle \emptyset, \psi \rangle \in R' \cup R''$ and substitution σ on $C' \cup C'' \subseteq C$. In the first case, clearly $\Gamma \vdash_H \varphi$. In the second case, since $\emptyset \vdash_H \sigma(\psi)$, we can present a derivation of φ from Γ in H, that is, $\Gamma \vdash_H \varphi$.

Step: Assume that φ has a derivation of length $n+1$ from Γ in $H' \cup H''$. The interesting case is $\varphi = \sigma(\psi)$ for some rule $\langle \Delta, \psi \rangle \in R' \cup R''$ and substitution σ on $C' \cup C'' \subseteq C$ where the formulas in $\sigma(\Delta)$ occur previously in the derivation. By the induction hypothesis, every formula in $\sigma(\Delta)$ has a derivation in H from Γ. Hence, $\sigma(\Delta) \subseteq \Gamma^{\vdash_H}$ and, by monotonicity, $(\sigma(\Delta))^{\vdash_H} \subseteq (\Gamma^{\vdash_H})^{\vdash_H}$. Moreover, $\sigma(\Delta) \vdash_H \sigma(\psi)$, that is, $\sigma(\psi) \in (\sigma(\Delta))^{\vdash_H}$. Hence, $\sigma(\psi) \in (\Gamma^{\vdash_H})^{\vdash_H}$ and, by idempotence, $\sigma(\psi) \in \Gamma^{\vdash_H}$, that is, $\Gamma \vdash_H \varphi$. $\triangleleft$

Propositions 2.2.19 and 2.2.20 show that $H' \cup H''$ is the supremum of H' and H'' with respect to the weakness ordering.

We synthesize the properties of fibring of Hilbert calculi as follows:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fibring;
- *canonical combination of logics*: minimal construction as we will see below.

Observe that fusion of logics is a particular case of fibring. Therefore, fusion of modal logics is also a universal construction.

Remark 2.2.21 Fibring of Hilbert calculi can be characterized as an universal construction.

In fact, recall the signature morphisms presented in Remark 2.1.10. Each signature morphism $h : C \to C'$ can be extended as expected to $h^* : L(C)) \to L(C')$, by defining $h^*(\xi) = \xi$ for every $\xi \in \Xi$. For convenience we frequently write h for denoting the extension h^* of h.

A Hilbert calculus morphism

$$h : \langle C, R \rangle \to \langle C', R' \rangle$$

is a signature morphism $h : C \to C'$ such that $h(\varphi)$ is derivable from $h(\Delta)$ in $\langle C', R' \rangle$, for every inference rule $\langle \Delta, \varphi \rangle$ in R. A weaker possibility for the notion of Hilbert calculus morphism would be to say that $\langle h(\Delta), h(\varphi) \rangle \in R'$ for every $\langle \Delta, \varphi \rangle \in R$. This weaker version is the more adequate in Chapter 7.

Recursive Hilbert calculi and their morphisms, with composition and identities as in **Sig**, constitute the category **Hil**. The category **Hil** is (finitely) cocomplete.

The fibring $H' \cup H''$ of $H' = \langle C', R' \rangle$ and $H'' = \langle C'', R'' \rangle$ is a pushout of the morphisms $h' : \langle C' \cap C'', \emptyset \rangle \to \langle C', R' \rangle$ and $h'' : \langle C' \cap C'', \emptyset \rangle \to \langle C'', R'' \rangle$, where h' and h'' are the signature inclusion morphisms. Figure 2.3 describes this situation.

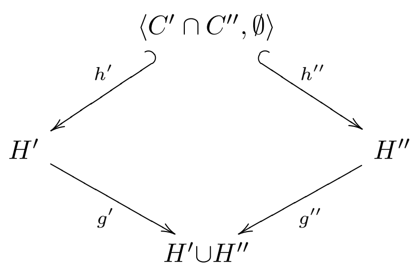


Figure 2.3: Fibring of Hilbert calculi as a pushout in **Hil**

The particular case of unconstrained fibring just corresponds to the coproduct of H' and H''. ∇

2.3 Preservation results

One of the important aspects of fibring is the possibility of obtaining transference results from the logics to be fibred to the logic resulting from the fibring. Herein, we study the preservation by fibring of some metatheorems (for instance the metatheorem of deduction) as well as the preservation of interpolation.

2.3.1 Global and local derivation

We start by investigating the preservation of derivation by fibring. The results that we obtain here will be used in the sequel.

In order to formulate the preservation results, we need to refine the notion of derivation so that we can distinguish between global and local derivations. The distinction is very clear when dealing, for instance, with modal logic. Although this distinction is better motivated using semantic arguments, to be discussed in the next chapter, it is still possible to motivate it using Hilbert calculi.

Consider, as motivating example, a Hilbert calculus H with a connective $\Rightarrow$ having the properties of the usual implication. Assuming that $\Gamma, \psi \vdash_H \varphi$ holds, it may be the case that $\Gamma \vdash_H (\psi \Rightarrow \varphi)$ holds in H or not. We usually say that the calculus H has the metatheorem of deduction (MTD) whenever $\Gamma \vdash_H (\psi \Rightarrow \varphi)$ always follows from $\Gamma, \psi \vdash_H \varphi$. For example, it is well known that propositional classical logic has MTD. On the other hand, in modal logic, $\psi \vdash_H (\Box\psi)$ holds but, in general, it not the case that $\vdash_H (\psi \Rightarrow (\Box\psi))$. Also, in first order classical logic, $\psi \vdash_H (\forall x\, \psi)$ holds but, in general, it not the case that $\vdash_H (\psi \Rightarrow (\forall x\, \psi))$. However, if ψ is a theorem of modal logic (that is, $\vdash_H \psi$), $\vdash_H (\psi \Rightarrow (\Box\psi))$ follows from $\psi \vdash_H (\Box\psi)$. Also, if there exists a derivation of φ from $\Gamma \cup \{\psi\}$ without using the necessitation rule then $\Gamma \vdash_H (\psi \Rightarrow \varphi)$ holds. Similar arguments can be applied to first order classical logic.

These examples suggest that two kinds of derivations can be recognized. In the first one, all the inference rules can be applied freely but, as a consequence, *MTD* can be lost. This kind of derivations will be called *global* derivations. In the second one, in order to guarantee *MTD*, some inference rules can be applied freely but other can only be applied to theorems. This kind of derivations will be called *local* derivations. For instance, in modal logic, the rule of modus ponens can be applied freely, but the necessitation rule can only be applied to theorems in order to guarantee *MTD*. The same holds in first order classical logic with respect to the generalization rule.

When the distinction between global and local derivations is made, it is possible to give a formulation of the local metatheorem of deduction without any provisos. In modal logic people usually work in this way. In first-order logic, people use to work with global reasoning and with a global metatheorem of deduction having provisos. In Subsection 2.3.3 we will see that for global reasoning in modal logic one can give a generalized version of the metatheorem of deduction.

We say that a logic supporting both local and global reasoning has *careful reasoning*. For more details on the interest of the distinction see also Subsection 2.3.3.

Although these motivating examples are based on implication and *MTD*, the idea of global and local derivations can be generalized. This motivates the following definition.

Definition 2.3.1 A *Hilbert calculus with careful reasoning* is a triple

$$H = \langle C, R_g, R_\ell \rangle$$

where C is a signature, $R_g \cup R_\ell$ is a set of inference rules over C such that $R_\ell \subseteq R_g$ and $\Delta \neq \emptyset$ for each $\langle \Delta, \varphi \rangle \in R_g \backslash R_\ell$. ∇

The notion of recursive Hilbert calculus, presented in 2.2.12, is easily extended to Hilbert calculus with careful reasoning.

The elements of R_g are the global inference rules and the elements of R_ℓ are the local inference rules. Note that each local rule is also a global rule but we can have global rules that are not local rules. From this point on we will work with Hilbert calculus with careful reasoning and call it just Hilbert calculus.

In modal logic, it is usual to distinguish between local and global reasoning (see [167]) namely because of the metatheorem of deduction that holds for local reasoning but not for global reasoning. In first-order logic most people work with global reasoning and give a constrained version of the metatheorem of deduction.

We can now state when a formula is globally derivable and when it is locally derivable.

Definition 2.3.2 Let $H = \langle C, R_g, R_\ell \rangle$ be a Hilbert calculus.

A formula $\varphi \in L(C)$ is *globally derivable* from $\Gamma \subseteq L(C)$ in H if there is a finite sequence $\varphi_1 \dots \varphi_n$ of formulas such that:

- φ_n is φ;
- for $i = 1, \dots, n$, each φ_i is either an element of Γ, or there exist a substitution σ and a rule $\langle \Delta, \psi \rangle \in R_g$ such that $\sigma(\Delta) \subseteq \{\varphi_1, \dots, \varphi_{i-1}\}$ and φ_i is $\sigma(\psi)$.

The sequence $\varphi_1 \dots \varphi_n$ is a *global derivation* of φ from Γ and we write

$$\Gamma \vdash^g_H \varphi$$

to denote that φ is globally derivable from Γ in H.

In a global derivation we can freely use all the rules. In a local derivation we can only use the global rules providing that the premises are theorems (that is were obtained without using the hypotheses).

A formula $\varphi \in L(C)$ is *locally derivable* from $\Gamma \subseteq L(C)$ in H if there is a finite sequence $\varphi_1 \dots \varphi_n$ of formulas such that:

- φ_n is φ;
- for $i = 1, \dots, n$, each φ_i is either an element of Γ, or $\vdash^g_H \varphi$, or there exist a substitution σ and an inference rule $\langle \Delta, \psi \rangle \in R_\ell$ such that $\sigma(\Delta) \subseteq \{\varphi_1, \dots, \varphi_{i-1}\}$ and φ_i is $\sigma(\psi)$.

The sequence $\varphi_1 \dots \varphi_n$ is a *local derivation* of φ from Γ and we write

$$\Gamma \vdash^\ell_H \varphi$$

to denote that φ is locally derivable from Γ in H. ∇

Any inference rule can be used in a global derivation. Local derivations only use local inference rules but they can also use global theorems, that is, formulas globally derivable from the empty set. Note that every formula that is locally derivable from a set Γ is also globally derivable from Γ. Moreover, every formula that is globally derivable from the empty set is also locally derivable from the empty set. That is, local theorems are also global theorems and vice versa.

We denote by $\Gamma^{\vdash^g_H}$ and $\Gamma^{\vdash^\ell_H}$ respectively the set of formulas globally derivable from Γ in H and the set of formulas locally derivable from Γ in H. The set Γ is said to be *globally closed* if $\Gamma^{\vdash^g_H} = \Gamma$.

As an illustration we consider again modal logic with these two levels of reasoning.

Example 2.3.3 Recall the modal signature introduced in Example 2.1.4. A Hilbert calculus with careful reasoning for propositional normal modal logic **K** includes, besides that signature, the set R_ℓ with the following local inferences rules

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1))\rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3)))\rangle$;
- $\langle \emptyset, (((\neg\, \xi_1) \Rightarrow (\neg\, \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1))\rangle$
- $\langle \emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2)))\rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$;

and the set R_g with the following global inferences rules

- all the inference rules in R_ℓ;
- $\langle \{\xi_1\}, (\Box\xi_1)\rangle$.

This Hilbert calculus has exactly the same inference rules as the Hilbert calculus for propositional normal modal logic **K** presented in Example 2.2.4, but herein rules can be global or local (or both). All the inference rules presented before are now global inference rules. All the inference rules are also local inference rules with the exception of the necessitation rule Nec. Hence, Nec is the only global rule that is not a local rule. As a consequence, $\xi_1 \vdash_H^g (\Box\xi_1)$ but $\xi_1 \vdash_H^\ell (\Box\xi_1)$ does not hold because we would be applying a global rule to an hypothesis.

A Hilbert calculus for modal logics **S4** is similar to the above Hilbert calculus but also includes the axioms T and 4 as local rules. Similarly with respect to modal logic **B** where B is the new axiom. ∇

Example 2.3.4 A Hilbert calculus for classical logic is $\langle C, R_g, R_\ell\rangle$ where C is the usual classical signature, R_g is the set of inference rules in Example 2.2.3 and $R_\ell = R_g$.

Hilbert calculus for the intuitionistic logic, the 3-valued Gödel logic and the 3-valued Łukasiewicz logic are obtained in a similar way considering the set of inference rules in respectively Examples 2.2.6, 2.2.7 and 2.2.8. ∇

Remark 2.3.5 Note that a Hilbert calculus $H = \langle C, R_g, R_\ell\rangle$ induces two compact and structural consequence systems, the global consequence system $\langle C, \vdash_H^g\rangle$ and the local consequence system $\langle C, \vdash_H^\ell\rangle$. The proof is similar to the one presented for Proposition 2.2.11. Observe that $\langle C, \vdash_H^g\rangle$ is just the consequence system referred to therein. It is easy to see that

$$\langle C, \vdash_H^\ell\rangle \leq \langle C, \vdash_H^g\rangle.$$

The weakness relation between Hilbert calculi extends as expected to calculi involving global and local inference rules. We say that H is weaker than H' (or H' is stronger than H), written $H \leq H'$, whenever $C \leq C'$, $\Gamma^{\vdash^g_H} \subseteq \Gamma^{\vdash^g_{H'}}$ and $\Gamma^{\vdash^\ell_H} \subseteq \Gamma^{\vdash^\ell_{H'}}$, for every $\Gamma \subseteq L(C)$.

Fibring also extends to these Hilbert calculi as expected: the Hilbert calculus

$$H' \cup H'' = \langle C' \cup C'', {R_g}' \cup {R_g}'', {R_\ell}' \cup {R_\ell}'' \rangle$$

is the fibring of $H' = \langle C', {R_g}', {R_\ell}' \rangle$ and $H'' = \langle C'', {R_g}'', {R_\ell}'' \rangle$. It is straightforward to conclude that

$$H' \leq H' \cup H'' \text{ and } H'' \leq H' \cup H''.$$

Clearly recursiveness of Hilbert calculi is preserved by fibring. ∇

Proposition 2.3.6 *Fibring preserves global derivations and local derivations.*

In the sequel, we assume that Hilbert calculi include a set of global inference rules and a set of local inference rules.

2.3.2 Metatheorems

Some metatheorems, such as the metatheorem of modus ponens and the metatheorem of deduction, are important features of several logics. It is relevant to know if those metatheorems are preserved by fibring. In [282] some metatheorem preservation results are presented.

Herein, we establish sufficient conditions for the preservation of several metatheorems by fibring. Since these results only involve syntactic arguments we present them in the Hilbert calculus setting.

In the sequel, whenever we want to state a condition such as "if statements *stm*$_1$,..., *stm*$_n$ hold then statement *stm* also holds", for simplicity, we may just use the notation

$$\frac{stm_1 \ldots stm_n}{stm}$$

We may also have more than one statement below the horizontal line meaning that providing that the statements of the numerator are true then the statements of the denominator are also true.

We first consider the preservation of the metatheorem of modus ponens. To begin with, we have to define the notion of metatheorem of modus ponens in an arbitrary Hilbert calculus.

Definition 2.3.7 A Hilbert calculus H has the *metatheorem of modus ponens* (*MTMP*) if there is a binary connective $\Rightarrow$ such that

$$\frac{\Gamma \vdash^\ell_H (\varphi_1 \Rightarrow \varphi_2)}{\Gamma, \varphi_1 \vdash^\ell_H \varphi_2} \qquad (MTMP)$$

for every set of formulas Γ and formulas φ_1, φ_2. ▽

We observe that $\Rightarrow$ can be either a connective in the signature or it can be a derived one. For example, in classical propositional logic we can take $\Rightarrow$ as the implication. But we can also define $(\varphi_1 \Rightarrow \varphi_2)$ as $((\neg\, \varphi_1) \vee \varphi_2)$ assuming that $\neg$ and $\vee$ are the connectives in the signature.

It is possible to give a necessary and sufficient condition for a Hilbert calculus to have *MTMP* which does not involve meta-reasoning.

Proposition 2.3.8 *The MTMP holds in a Hilbert calculus H if and only if*

- $(\xi_1 \Rightarrow \xi_2), \xi_1 \vdash^{\ell}_{H} \xi_2$.

Proof. Assume that *MTMP* holds and let Γ be $\{(\xi_1 \Rightarrow \xi_2)\}$ and let φ_1, φ_2 be respectively ξ_1, ξ_2. Since $(\xi_1 \Rightarrow \xi_2) \vdash^{\ell}_{H} (\xi_1 \Rightarrow \xi_2)$, from *MTMP* it follows that $(\xi_1 \Rightarrow \xi_2), \xi_1 \vdash^{\ell}_{H} \xi_2$.

Conversely, assume $(\xi_1 \Rightarrow \xi_2), \xi_1 \vdash^{\ell}_{H} \xi_2$ and suppose that $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$. Then, by monotonicity, also $\Gamma, \varphi_1 \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$. Consider the substitution σ such that $\sigma(\xi_i) = \varphi_i$, $i = 1, 2$. Using structurality, it follows that $(\varphi_1 \Rightarrow \varphi_2), \varphi_1 \vdash^{\ell}_{H} \varphi_2$. Hence, $\Gamma, \varphi_1 \vdash^{\ell}_{H} \varphi_2$ using idempotence. ◁

Example 2.3.9 Recall that all the Hilbert calculi presented in Section 2.2 have the modus ponens rule. Since this rule is a local rule when we refine these Hilbert calculi, see Examples 2.3.3 and 2.3.4, we conclude that all of them have *MTMP*. ▽

Fibring preserves *MTMP* even if only one of the components has *MTMP*.

Theorem 2.3.10 *The fibring of two Hilbert calculi has MTMP provided that at least one of the Hilbert calculi has MTMP.*

Proof. Let H' and H'' be Hilbert calculi where H' has *MTMP*. From Proposition 2.3.8, $(\xi_1 \Rightarrow \xi_2), \xi_1 \vdash^{\ell}_{H_1} \xi_2$. Using Remark 2.3.5, $(\xi_1 \Rightarrow \xi_2), \xi_1 \vdash^{\ell}_{H' \cup H''} \xi_2$. Hence, using again Proposition 2.3.8, $H' \cup H''$ has *MTMP*. ◁

We now consider the metatheorem of deduction.

Definition 2.3.11 A Hilbert calculus H has the *metatheorem of deduction* (*MTD*) if there is a binary connective $\Rightarrow$ such that:

$$\frac{\Gamma, \varphi_1 \vdash^{\ell}_{H} \varphi_2}{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)} \qquad (MTD)$$

for every set of formulas Γ and formulas φ_1, φ_2. ▽

The definition above corresponds to a local version of the metatheorem of deduction. In Subsection 2.3.3 we will consider a more relaxed version of the metatheorem of deduction that will also cover global derivation. It is possible to give a sufficient condition for a Hilbert calculus to have *MTD* that is also a necessary condition if H also has *MTMP*.

Proposition 2.3.12 *A Hilbert calculus H has MTD if*

1. $\vdash^{\ell}_{H} (\xi_1 \Rightarrow \xi_1)$;
2. $\xi_1 \vdash^{\ell}_{H} (\xi_2 \Rightarrow \xi_1)$;
3. $(\xi \Rightarrow \delta_1), \ldots, (\xi \Rightarrow \delta_k) \vdash^{\ell}_{H} (\xi \Rightarrow \varphi)$ *for each local rule* $\langle\{\delta_1, \ldots, \delta_k\}, \varphi\rangle$ *where* $\xi \in \Xi$ *does not occur in any of the formulas of the rule.*

Conversely, if H has MTD then the conditions 1 and 2 above hold. If H also has MTMP with respect to the same connective $\Rightarrow$ then condition 3 also holds.

Proof. Assume that the three conditions hold and that $\Gamma, \varphi_1 \vdash^{\ell}_{H} \varphi_2$. We prove by induction on the length of a derivation of φ_2 from $\Gamma \cup \{\varphi_1\}$ that $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$.

Base: If φ_2 has a derivation of length one from $\Gamma \cup \{\varphi_1\}$ then either $\varphi_2 \in \Gamma \cup \{\varphi_1\}$, or $\vdash^{g}_{H} \varphi_2$, or $\varphi_2 = \sigma(\varphi)$ for some local axiom $\langle\emptyset, \varphi\rangle$ and substitution σ. In the first case, if φ_2 is φ_1, we consider a substitution σ such that $\sigma(\xi_1) = \varphi_1$ and use 1, structurality and monotonicity. If $\varphi_1 \in \Gamma$, we consider σ such that $\sigma(\xi_1) = \varphi_2$, $\sigma(\xi_2) = \varphi_1$ and use 2, structurality and monotonicity. In the second case, we have $\vdash^{\ell}_{H} \varphi_2$. Considering again a substitution σ such that $\sigma(\xi_1) = \varphi_2$, $\sigma(\xi_2) = \varphi_1$ and using 2 and structurality we have $\varphi_2 \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$. Hence, using idempotence and monotonicity, we get $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$. In the third case, we also have $\vdash^{\ell}_{H} \varphi_2$ and we reason as before.

Step: If φ_2 has a derivation of length $n + 1$, we reason as above if φ_2 is as in the base case. Otherwise, $\varphi_2 = \sigma(\varphi)$ for some local rule $\langle\{\delta_1, \ldots, \delta_k\}, \varphi\rangle$ and substitution σ such that $\sigma(\delta_1), \ldots, \sigma(\delta_k)$ occur previously in the derivation. Using the induction hypothesis, $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \sigma(\delta_i))$, $i = 1, \ldots, k$. Since ξ does not occur in the rule, we may consider without loss of generality that $\sigma(\xi) = \varphi_1$. Hence, using 3 and structurality, $\{(\varphi_1 \Rightarrow \sigma(\delta_1)), \ldots, (\varphi_1 \Rightarrow \sigma(\delta_k))\} \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$. Using idempotence, $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$.

Conversely, assume now that *MTD* holds. We get 1 letting Γ be $\emptyset$ and φ_1, φ_2 be ξ_1. We get 2 letting Γ be $\{\xi_1\}$ and φ_1, φ_2 be respectively ξ_2, ξ_1. Finally, let Γ be $\{(\xi \Rightarrow \delta_1), \ldots, (\xi \Rightarrow \delta_k)\}$ and φ_1, φ_2 be respectively ξ, φ. Since $\Gamma \vdash^{\ell}_{H} (\xi \Rightarrow \delta_1)$, from *MTMP* we conclude $\Gamma, \xi \vdash^{\ell}_{H} \delta_1$. Similarly, $\Gamma, \xi \vdash^{\ell}_{H} \delta_i$, $i = 2, \ldots, k$. Hence, using the rule, $\Gamma, \xi \vdash^{\ell}_{H} \varphi$ and, using *MTD*, we get 3. ◁

Example 2.3.13 The Hilbert calculus for classical logic referred to in Example 2.3.4 has *MTD*. In fact, using the first two axioms, $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1))\rangle$ and $\langle\emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3)))\rangle$, and the rule modus ponens, it is easy to prove that conditions 1 and 2 of Proposition 2.3.12 hold and that modus ponens satisfies condition

$$\{(\xi \Rightarrow \xi_1), (\xi \Rightarrow (\xi_1 \Rightarrow \xi_2))\} \vdash^{\ell}_{H} (\xi \Rightarrow \xi_2).$$

The Hilbert calculus for intuitionistic logic referred to in Example 2.3.4 also has the *MTD* for a similar reason. Similarly, the Hilbert calculus for the 3-valued Gödel logic also has *MTD*.

The Hilbert calculus for modal logic **K** in Example 2.3.3 also has *MTD* because the necessitation rule is not a local rule. Note that condition 3 of Proposition 2.3.12 does not hold for this rule. ▽

The metatheorem *MTD* is preserved by fibring, in the presence of *MTMP*.

Theorem 2.3.14 *The fibring of two Hilbert calculi has MTD provided that they have MTMP and MTD, and the binary connective* $\Rightarrow$ *in MTMP and MTD is shared.*

Proof. Let H' and H'' be Hilbert calculi having *MTMP* and *MTD*. From Proposition 2.3.12, conditions 1 and 2 of this Proposition hold, in particular, in H'. Using Remark 2.3.5, they also hold in the fibring $H' \cup H''$. Thus, these conditions hold in the fibring even if only one of the calculi has *MTD*. But condition 3 involves all local rules of $H' \cup H''$ and so we have to guarantee this condition for the rules of both calculi. Since H' and H'' have *MTD* and *MTMP*, condition 3 holds in both. Hence, using Remark 2.3.5 and taking into account that $\Rightarrow$ is shared and the local inference rules of $H' \cup H''$ are the local inference rules of H' and the ones of H'', condition 3 holds in the fibring. By Proposition 2.3.8, $H' \cup H''$ has *MTD*. ◁

As a particular case, observe that fusion preserves the *MTMP* and *MTD* taking the binary connective as the implication. This is a consequence of the fact that modal Hilbert calculi have the local version of the metatheorem of deduction.

The metatheorem *MTMP* and *MTD* characterize the behavior of the usual classical implication. That is why we say that a Hilbert calculus having *MTMP* and *MTD* has implication.

Definition 2.3.15 A Hilbert calculus H is said to have *implication* $\Rightarrow$ if $\Rightarrow$ is a binary connective satisfying *MTMP* and *MTD*. ▽

As we will see later on, preservation of *MTMP* and *MTD* is important when proving the preservation of completeness. Clearly, fibring of Hilbert calculi having implication also has implication if the implication connectives is shared.

Proposition 2.3.16 *The fibring of two Hilbert calculi having implication also has implication provided the implication connective is shared.*

Proof. Use Theorem 2.3.10 and Theorem 2.3.14. ◁

We now consider three other metatheorems, these ones related with the properties of equivalence and its relation with implication. The preservation of these metatheorems is also important for establishing the preservation of completeness. The first two are the metatheorems of biconditionaly 1 and 2.

Definition 2.3.17 A Hilbert calculus H has the *metatheorem of biconditional 1* (*MTB1*) if there are binary connectives $\Rightarrow$ and $\Leftrightarrow$ such that:

$$\frac{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2) \qquad \Gamma \vdash^{\ell}_{H} (\varphi_2 \Rightarrow \varphi_1)}{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Leftrightarrow \varphi_2)} \qquad (MTB1)$$

for every set of formulas Γ and formulas φ_1, φ_2.

A Hilbert calculus H has the *metatheorem of biconditional 2* (*MTB2*) if there are binary connectives $\Rightarrow$ and $\Leftrightarrow$ such that :

$$\frac{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Leftrightarrow \varphi_2)}{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2) \qquad \Gamma \vdash^{\ell}_{H} (\varphi_2 \Rightarrow \varphi_1)} \qquad (MTB2)$$

for every set of formulas Γ and formulas φ_1, φ_2. ∇

Note that if a Hilbert calculus has *MTB1* and *MTB2* then from $\Gamma \vdash^{\ell}_{H} (\varphi_1 \Leftrightarrow \varphi_2)$ we can also conclude $\Gamma \vdash^{\ell}_{H} (\varphi_2 \Leftrightarrow \varphi_1)$.

We now present necessary and sufficient conditions for a Hilbert calculus to have *MTB1* and *MTB2*.

Proposition 2.3.18 *The MTB1 holds in a Hilbert calculus H if and only if*

- $(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1) \vdash^{\ell}_{H} (\xi_1 \Leftrightarrow \xi_2)$.

Proof. Assume that *MTB1* holds and let Γ be $\{(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1)\}$ and φ_1, φ_2 be respectively ξ_1, ξ_2. Since

$$\{(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1)\} \vdash^{\ell}_{H} (\xi_1 \Rightarrow \xi_2)$$

and

$$\{(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1)\} \vdash^{\ell}_{H} (\xi_2 \Rightarrow \xi_1)$$

from *MTB1* we get

$$\{(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1)\} \vdash^{\ell}_{H} (\xi_1 \Leftrightarrow \xi_2).$$

Conversely, assume that $(\xi_1 \Rightarrow \xi_2), (\xi_2 \Rightarrow \xi_1) \vdash^{\ell}_{H} (\xi_1 \Leftrightarrow \xi_2)$ and suppose that

$$\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2)$$

and

$$\Gamma \vdash^{\ell}_{H} (\varphi_2 \Rightarrow \varphi_1).$$

Consider the substitution σ such that $\sigma(\xi_i) = \varphi_i$, $i = 1, 2$. Using structurality,

$$\{(\varphi_1 \Rightarrow \varphi_2), (\varphi_2 \Rightarrow \varphi_1)\} \vdash_H^\ell (\varphi_1 \Leftrightarrow \varphi_2).$$

Hence, $\Gamma \vdash_H^\ell (\varphi_1 \Leftrightarrow \varphi_2)$ and so *MTB1* holds in H. ◁

Proposition 2.3.19 *The MTB2 holds in a Hilbert calculus H if and only if*

1. $(\xi_1 \Leftrightarrow \xi_2) \vdash_H^\ell (\xi_1 \Rightarrow \xi_2)$;
2. $(\xi_1 \Leftrightarrow \xi_2) \vdash_H^\ell (\xi_2 \Rightarrow \xi_1)$.

Proof. The proof is similar to the one presented for Proposition 2.3.18, using the obvious suitable instantiations for Γ, φ_1 and φ_2 and suitable substitutions σ. ◁

Example 2.3.20 The connective $\Leftrightarrow$ is usually defined as an abbreviation:

$$(\varphi_1 \Leftrightarrow \varphi_2) =_{\text{def}} ((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)).$$

Consider the Hilbert calculus for intuitionistic logic referred to in Example 2.3.4. In order to cope with Definition 2.3.17, we need to add the symbol $\Leftrightarrow$ to the signature for intuitionistic logic considered in Definition 2.1.4. Moreover, we need to add to the corresponding Hilbert calculus the following two axioms:

- $\langle \emptyset, ((\xi_1 \Leftrightarrow \xi_2) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \wedge (\xi_2 \Rightarrow \xi_1))) \rangle$;
- $\langle \emptyset, (((\xi_1 \Rightarrow \xi_2) \wedge (\xi_2 \Rightarrow \xi_1)) \Rightarrow (\xi_1 \Leftrightarrow \xi_2)) \rangle$.

Using these axioms together with

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow (\xi_1 \wedge \xi_2))) \rangle$
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_1) \rangle$
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_2) \rangle$

and modus ponens, we easily get the conditions in Propositions 2.3.18 and 2.3.19. Hence, this new Hilbert calculus has *MTB1* and *MTB2*. Similarly, the Hilbert calculus for the 3-valued Gödel logic, extended with $\Leftrightarrow$ as above, also has *MTB1* and *MTB2*.

In what concerns classical logic, a similar technique can be applied, but now the axioms to be added to the corresponding Hilbert calculus are:

- $\langle \emptyset, ((\xi_1 \Leftrightarrow \xi_2) \Rightarrow (\neg((\xi_1 \Rightarrow \xi_2) \Rightarrow (\neg(\xi_2 \Rightarrow \xi_1))))) \rangle$;
- $\langle \emptyset, ((\neg((\xi_1 \Rightarrow \xi_2) \Rightarrow (\neg(\xi_2 \Rightarrow \xi_1)))) \Rightarrow (\xi_1 \Leftrightarrow \xi_2)) \rangle$.

Hence, this extended Hilbert calculus also has *MTB1* and *MTB2*. Clearly, similar remarks hold for modal logic **K** and for 3-valued Łuckasiewicz logic. ▽

Fibring also preserves $MTB1$ and $MTB2$.

Theorem 2.3.21 *The fibring of two Hilbert calculi has MTB1 provided that at least one of the Hilbert calculi has MTB1.*

Proof. The proof is similar to the one presented for Theorem 2.3.10. ◁

Theorem 2.3.22 *The fibring of two Hilbert calculi has MTB2 provided that at least one of the Hilbert calculi has MTB2.*

Proof. The proof is similar to the one presented for Theorem 2.3.10. ◁

Next we consider the metatheorem of substitution of equivalents.

Definition 2.3.23 A Hilbert calculus H with implication has the *metatheorem of substitution of equivalents* (*MTSE*) if there is a binary connective $\Leftrightarrow$ such that:

$$\frac{\Gamma \vdash^{\ell}_{H} (\delta_1 \Leftrightarrow \delta_2)}{\Gamma \vdash^{\ell}_{H} (\varphi \Leftrightarrow \varphi')} \qquad (MTSE)$$

for every globally closed set of formulas Γ and formulas $\delta_1, \delta_2, \varphi$, where φ' is obtained from φ by replacing one of more occurrences of δ_1 by δ_2. ▽

We now present necessary and sufficient conditions for ensuring *MTSE* in Hilbert calculus verifying certain conditions.

Proposition 2.3.24 *Let H be a Hilbert calculus with implication $\Rightarrow$ where MTB1 and MTB2 hold with respect to $\Rightarrow$ and $\Leftrightarrow$. The MTSE holds in H if and only if*

- $\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_H} \vdash^{\ell}_{H} ((c(\xi_2, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \ldots, \xi_{2k-1})))$

for every constructor c of arity $k > 0$ in the signature.

Proof. Assume that *MTSE* holds and let

$$\Theta = \{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_H}.$$

Clearly $\Theta \vdash^{\ell}_{H} (\xi_{2i} \Leftrightarrow \xi_{2i-1})$ for $i = 1, \ldots, k$. Taking $i = 1$ and using *MTSE*, we get

$$\Theta \vdash^{\ell}_{H} ((c(\xi_2, \xi_4, \xi_6, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \xi_4, \xi_6, \ldots, \xi_{2k}))).$$

Taking $i = 2$ and using *MTSE*, we get

$$\Theta \vdash^{\ell}_{H} ((c(\xi_1, \xi_4, \xi_6, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \xi_3, \xi_6, \ldots, \xi_{2k}))).$$

Since H has *MTMP*, *MTD*, *MTB1* and *MTB2*, using these metatheorems we can conclude that

$$\Theta \vdash^{\ell}_{H} ((c(\xi_2, \xi_4, \xi_6, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \xi_3, \xi_6, \ldots, \xi_{2k}))).$$

Keeping on reasoning in this way, at the end we get

$$\Theta \vdash^{\ell}_{H} ((c(\xi_2,\ldots,\xi_{2k})) \Leftrightarrow (c(\xi_1,\ldots,\xi_{2k-1}))).$$

Conversely, assume that $\Theta \vdash^{\ell}_{H} ((c(\xi_2,\ldots,\xi_{2k}))\Leftrightarrow(c(\xi_1,\ldots,\xi_{2k-1})))$, for every c of arity $k>0$, and suppose that $\Gamma \vdash^{\ell}_{H} (\delta_1 \Leftrightarrow \delta_2)$. We prove by structural induction that

$$\Gamma \vdash^{\ell}_{H} (\varphi \Leftrightarrow \varphi')$$

where φ' is obtained from φ by replacing one or more occurrences of δ_1 by δ_2.

Base: If φ is a constructor of arity 0, $\varphi' = \varphi$. Since H has implication and *MTB1*, $\vdash^{\ell}_{H} (\xi_1 \Rightarrow \xi_1)$ hence, from *MTB1*, $\vdash^{\ell}_{H} (\xi_1 \Leftrightarrow \xi_1)$. Using structurality with σ such that

$$\sigma(\xi_1) = \varphi \quad \text{and} \quad \vdash^{\ell}_{H} (\varphi \Leftrightarrow \varphi)$$

hence, by monotonicity, $\Gamma \vdash^{\ell}_{H} (\varphi \Leftrightarrow \varphi')$. If $\varphi = \xi \in \Xi$ and $\xi \neq \delta_1$, $\varphi' = \varphi$ and we reason as before. If $\varphi = \xi = \delta_1$, $\varphi' = \delta_2$, hence $(\varphi \Leftrightarrow \varphi')$ is just $(\delta_1 \Leftrightarrow \delta_2)$ and we are done.

Step: Assume that φ is of the form $(c(\psi_1,\ldots,\psi_k))$. By the induction hypothesis, $\Gamma \vdash^{\ell}_{H} (\psi_i \Leftrightarrow \psi'_i)$ for $i=1,\ldots,k$, where ψ' is obtained from ψ by replacing one or more occurrences of δ_1 by δ_2. Thus also $\Gamma \vdash^{g}_{H} (\psi_i \Leftrightarrow \psi'_i)$ for $i=1,\ldots,k$. Since Γ is globally closed,

$$\{(\psi_i \Leftrightarrow \psi'_i) : i=1,\ldots,k\}^{\vdash^{g}_{H}} \subseteq \Gamma.$$

From the hypothesis,

$$\Theta \vdash^{\ell}_{H} ((c(\xi_2,\ldots,\xi_{2k}) \Leftrightarrow c(\xi_1,\ldots,\xi_{2k-1}))).$$

Using structurality and σ such that $\sigma(\xi_{2i}) = \psi_i$ and $\sigma(\xi_{2i-1}) = \psi'_i$, for $i=1,\ldots,k$, we get

$$\sigma(\Theta) \vdash^{\ell}_{H} ((c(\psi_1,\ldots,\psi_k)) \Leftrightarrow (c(\psi'_1,\ldots,\psi'_k))).$$

Since $\sigma(\Theta) \subseteq \{(\psi_i \Leftrightarrow \psi'_i) : i=1,\ldots,k\}^{\vdash^{g}_{H}} \subseteq \Gamma$, by monotonicity,

$$\Gamma \vdash^{\ell}_{H} ((c(\psi_1,\ldots,\psi_k)) \Leftrightarrow (c(\psi'_1,\ldots,\psi'_k))).$$

If we do not want to replace occurrences of δ_1 in some ψ_i we just use $\Gamma \vdash^{\ell}_{H} (\psi_i \Leftrightarrow \psi_i)$ that can be obtained as above. $\lhd$

It is worthwhile noticing that we could have defined *MTSE* without requiring Γ to be globally closed, and Proposition 2.3.24 would also hold without this requirement. However, as explained below, without this requirement, modal logics would not have *MTSE*. But, as we will see later on, this metatheorem plays an important role in our proof of completeness preservation and, naturally, we want this preservation result to hold in modal logics.

Example 2.3.25 Using the inference rules of the Hilbert calculus H for classical logic referred to in Example 2.3.4, we can prove that

$$\{(\xi_1 \Leftrightarrow \xi_2)\}^{\vdash^g_H} \vdash^\ell_H ((\neg\,\xi_1) \Leftrightarrow (\neg\,\xi_2))$$

and

$$\{(\xi_1 \Leftrightarrow \xi_2), (\xi_3 \Leftrightarrow \xi_4)\}^{\vdash^g_H} \vdash^\ell_H ((\xi_1 \Rightarrow \xi_3) \Leftrightarrow (\xi_2 \Rightarrow \xi_4)).$$

Since H has implication, *MTB1* and *MTB2*, the calculus has *MTSE*.

The Hilbert calculus for intuitionistic logic referred to in Example 2.3.4 also has *MTSE*. Note that herein we have to consider, besides $\Rightarrow$, the constructors $\wedge$ and $\vee$. Similarly, the Hilbert calculus for the 3-valued Gödel logic also has *MTSE*.

The Hilbert calculus for modal logic $\mathbf{K}$ in Example 2.3.3 also has *MTSE*. Thus, besides $\neg$ and $\Rightarrow$, we have to consider the constructor $\Box$. To prove

$$\{(\xi_1 \Leftrightarrow \xi_2)\}^{\vdash^g_H} \vdash^\ell_H ((\Box\xi_1) \Leftrightarrow (\Box\xi_2))$$

it is essential to consider the global closure of $\{(\xi_1 \Leftrightarrow \xi_2)\}$. ∇

Fibring preserves the metatheorem of substitution of equivalents when some conditions hold.

Theorem 2.3.26 *The fibring of two Hilbert calculi has MTSE provided that both Hilbert calculi have implication* $\Rightarrow$*, MTSE, MTB1, MTB2 and the binary connectives* $\Rightarrow$ *and* $\Leftrightarrow$ *in these metatheorems are shared.*

Proof. Let H', H'' be Hilbert calculi fulfilling the conditions above. By Proposition 2.3.24,

$$\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'}} \vdash^\ell_{H'} ((c(\xi_2, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \ldots, \xi_{2k-1})))$$

for every constructor c of arity $k > 0$ in the signature C' of H'. By Remark 2.3.5,

$$\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'}} \vdash^\ell_{H'\cup H''} ((c(\xi_2, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \ldots, \xi_{2k-1})))$$

and

$$\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'}} \subseteq \{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'\cup H''}}.$$

By monotonicity,

$$\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'\cup H''}} \vdash^\ell_{H'\cup H''} ((c(\xi_2, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \ldots, \xi_{2k-1})))$$

for every constructor c of arity $k > 0$ in the signature C' of H'. Similarly, and taking into account that $\Leftrightarrow$ is shared,

$$\{(\xi_{2i} \Leftrightarrow \xi_{2i-1}) : i = 1, \ldots, k\}^{\vdash^g_{H'\cup H''}} \vdash^\ell_{H'\cup H''} ((c(\xi_2, \ldots, \xi_{2k})) \Leftrightarrow (c(\xi_1, \ldots, \xi_{2k-1})))$$

for every constructor c of arity $k > 0$ in the signature C'' of H''. Hence, this result holds for every constructor c of arity $k > 0$ in the signature $C' \cup C''$ of $H' \cup H''$. By Proposition 2.3.16 and Theorems 2.3.21 and 2.3.22, $H' \cup H''$ has implication, *MTB1* and *MTB2*. By Proposition 2.3.24, $H' \cup H''$ has *MTSE*. $\triangleleft$

The metatheorems *MTB1*, *MTB2* and *MTSE* characterize the behavior of the usual classical equivalence. Hence, we say that a Hilbert calculus having *MTB1*, *MTB2* and *MTSE* has equivalence.

Definition 2.3.27 A Hilbert calculus H with implication $\Rightarrow$ has *equivalence* $\Leftrightarrow$ if $\Rightarrow$ and the binary connective $\Leftrightarrow$ satisfy *MTB1*, *MTB2* and *MTSE.* ∇

The next result is an obvious consequence of the previous ones.

Theorem 2.3.28 *The fibring of two Hilbert calculi having implication* $\Rightarrow$ *and equivalence* $\Leftrightarrow$ *also has implication and equivalence provided that* $\Rightarrow$ *and* $\Leftrightarrow$ *are shared.*

Proof. Use Proposition 2.3.16 and Theorems 2.3.21, 2.3.22 and 2.3.26. $\triangleleft$

Finally we consider the metatheorem of congruence.

Definition 2.3.29 A Hilbert calculus H has the *metatheorem of congruence* (*MTC*) if

$$\frac{\begin{array}{c}\Gamma, \varphi_i \vdash^{\ell}_{H} \psi_i \\ \Gamma, \psi_i \vdash^{\ell}_{H} \varphi_i\end{array} \quad i = 1, \dots, k}{\Gamma, (c(\varphi_1, \dots, \varphi_k)) \vdash^{\ell}_{H} (c(\psi_1, \dots, \psi_k))} \qquad (MTC)$$

for every globally closed set of formulas Γ, constructor c of arity $k > 0$ and formulas $\varphi_1, \dots, \varphi_k, \psi_1, \dots, \psi_k$. ∇

We now present a sufficient condition for a Hilbert calculus to have *MTC*.

Proposition 2.3.30 *Any Hilbert calculus* H *with implication and equivalence has MTC.*

Proof. Assume that $\Gamma, \varphi_i \vdash^{\ell}_{H} \psi_i$ and $\Gamma, \psi_i \vdash^{\ell}_{H} \varphi_i$ for $i = 1, \dots, k$, where Γ is globally closed. Since H has *MTD*, $\Gamma \vdash^{\ell}_{H} (\varphi_i \Rightarrow \psi_i)$ and $\Gamma \vdash^{\ell}_{H} (\psi_i \Rightarrow \varphi_i)$ for $i = 1, \dots, k$. Since H has *MTB1*, $\Gamma \vdash^{\ell}_{H} (\varphi_i \Leftrightarrow \psi_i)$ for $i = 1, \dots, k$. Since H has *MTSE*, in particular,

$$\Gamma \vdash^{\ell}_{H} ((c(\varphi_1, \varphi_2, \dots, \varphi_k)) \Leftrightarrow (c(\psi_1, \varphi_2, \dots, \varphi_k)))$$

and

$$\Gamma \vdash^{\ell}_{H} ((c(\psi_1, \varphi_2, \dots, \varphi_k)) \Leftrightarrow (c(\psi_1, \psi_2, \dots, \varphi_k))).$$

Since H has *MTMP*, *MTD*, *MTB1* and *MTB2*, we can prove that

$$\Gamma \vdash^{\ell}_{H} ((c(\varphi_1, \varphi_2, \dots, \varphi_k)) \Leftrightarrow (c(\psi_1, \psi_2, \dots, \varphi_k))).$$

Keeping on using $\Gamma \vdash^{\ell}_{H} (\psi_i \Rightarrow \varphi_i)$ for $i > 2$, at the end we get

$$\Gamma \vdash^{\ell}_{H} ((c(\varphi_1, \varphi_2, \dots, \varphi_k)) \Leftrightarrow (c(\psi_1, \psi_2, \dots, \psi_k))).$$

Since H has *MTB2*,

$$\Gamma \vdash^{\ell}_{H} ((c(\varphi_1, \varphi_2, \dots, \varphi_k)) \Rightarrow (c(\psi_1, \psi_2, \dots, \psi_k))).$$

Since H has *MTMP*, $\Gamma, (c(\varphi_1, \varphi_2, \dots, \varphi_k)) \vdash^{\ell}_{H} (c(\psi_1, \psi_2, \dots, \psi_k))$. ◁

The metatheorem *MTC* could also have been defined without requiring Γ to be globally closed. The reason why this requirement is considered is similar to the one explained above for *MTSE*.

Example 2.3.31 The Hilbert calculi for the classical, intuitionistic and Gödel logics presented in Example 2.3.4 and the modal logics in Example 2.3.3 all have *MTC*. ▽

The metatheorem of congruence is not always preserved by fibring. Consider the fibring of two Hilbert calculi H', H'' with the following signatures and rules:

$$C'_0 = \{p_1, p_2, p_3\} \quad C'_1 = \{c\} \quad C'_k = \emptyset \text{ for } k > 1$$
$$R_\ell{}' = \emptyset \quad R_g{}' = \{\langle\{\xi\}, c(\xi)\rangle\}$$
$$C''_0 = \{p_1, p_2, p_3\} \quad C''_k = \emptyset \text{ for } k > 0$$
$$R_\ell{}'' = R_g{}'' = \{\langle\{p_1, p_2\}, p_3\rangle, \langle\{p_1, p_3\}, p_2\rangle\}$$

Both H' and H'' are congruent, but their fibring $H' \cup H''$ is not congruent. In fact, considering

$$\Gamma = \{p_1\}^{\vdash^g_H} = \{c^n(p_1) : n \geq 0\},$$

from Γ, the formulas p_2 and p_3 are locally interderivable in $H' \cup H''$ but $c(p_2)$ and $c(p_3)$ are not. It seems that the problem comes from the fact that H'' does not have schematic rules.

However, it is possible to establish a sufficient condition for the preservation of congruence by fibring. Since equivalence is preserved by fibring, *MTC* is preserved by fibring of Hilbert calculi having equivalence and sharing implication and equivalence.

Theorem 2.3.32 *The fibring of Hilbert calculi with implication and equivalence has MTC provided that the implication and the equivalence connectives are shared.*

Proof. Use Proposition 2.3.30 and Theorem 2.3.28. ◁

Contrarily to fibring, observe that, in this context, fusion has always the metatheorem of congruence.

2.3.3 Interpolation

What is now generally known as Craig interpolation is a heritage of the seminal results proved by William Craig [65] in a proof-theoretic context for first-order logic. Several abstractions have been considered either in a proof-theoretical vein (e.g. [44, 43]) or in (non-constructive) model-theoretical style (e.g. for modal and positive logics as in [191, 192], for intuitionistic logic as in [102] and for hybrid logics as in [7, 8]).

The importance of Craig interpolation for some fundamental problems of complexity theory is analyzed in [213] and further developed in [214]. Interpolation has recently acquired practical relevance in engineering applications, namely, when formality and modularity are invoked [20], in software model-checking as in [147], and in SAT-based methods of unbounded symbolic model-checking as in [204].

The main objective of this section is to investigate preservation of both global and local interpolation. Hence we are going to work with careful reasoning Hilbert calculi. In this way, we will also relate global and local interpolation. The metatheorem of deduction will play an essential role providing that we work with a generalization of the one presented in Subsection 2.3.2.

In the previous sections, we have considered a fixed set Ξ of schema variables (recall Remark 2.1.3). Herein, however, we may need different sets of schema variables. Hence, from now on, we use $L(C,\Xi')$ to denote the language over C, as in Definition 2.1.5, but considering the set of schema variables Ξ'.

In the sequel, we use d when we want to refer to either local or global reasoning. We will use the notation

$$\Gamma \vdash^d_{H,\Xi'} \varphi$$

to denote that there exists a derivation of φ from Γ in H in which only formulas in $L(C,\Xi')$ occur. Moreover,

$$(\Gamma)^{\vdash^d_{H,\Xi'}}$$

will denote the set $\{\varphi \in L(C,\Xi') : \Gamma \vdash^d_{H,\Xi'} \varphi\}$ for $\Gamma \subseteq L(C,\Xi')$.

For our investigation we need a generalized version of the metatheorem of deduction.

Definition 2.3.33 Let H be a Hilbert calculus.

(i) H has the *d-metatheorem of deduction* (*d-MTD*) if there is a finite set of formulas $\Delta \subseteq L(C,\{\xi_1,\xi_2\})$ such that, for every $\Gamma \cup \{\varphi_1,\varphi_2\} \subseteq L(C,\Xi)$:

$$\text{if } \Gamma,\varphi_1 \vdash^d_{H,\Xi} \varphi_2 \text{ then } \Gamma \vdash^d_{H,\Xi} \Delta(\varphi_1,\varphi_2)$$

where $\Delta(\varphi_1,\varphi_2)$ is obtained from Δ by substituting ξ_i by φ_i for $i = 1,2$.

(ii) H has the *d-metatheorem of modus ponens* (*d-MTMP*) if there is a finite set of formulas $\Delta \subseteq L(C,\{\xi_1,\xi_2\})$ such that, for every $\Gamma \cup \{\varphi_1,\varphi_2\} \subseteq L(C,\Xi)$:

$$\text{if } \Gamma \vdash^d_{H,\Xi} \Delta(\varphi_1,\varphi_2) \text{ then } \Gamma,\varphi_1 \vdash^d_{H,\Xi} \varphi_2$$

where $\Delta(\varphi_1,\varphi_2)$ is obtained from Δ by substituting ξ_i by φ_i for $i = 1,2$.

We may refer to Δ as the *base set.* ▽

Example 2.3.34 For instance:

- Classical propositional logic has *g-MTD*, *g-MTMP*, *ℓ-MTD* and *ℓ-MTMP* taking $\Delta = \{(\xi_1 \Rightarrow \xi_2)\}$.
- Modal logics have *ℓ-MTD* and *ℓ-MTMP* with base set $\Delta = \{(\xi_1 \Rightarrow \xi_2)\}$.
- Modal logic in general does not have *g-MTD*. Modal logic **K4** has *g-MTD* and *g-MTMP* taking $\Delta = \{(((\Box\xi_1) \wedge \xi_1) \Rightarrow \xi_2)\}$. Modal logic **S4** has *g-MTD* and *g-MTMP* taking $\Delta = \{((\Box\xi_1) \Rightarrow \xi_2)\}$.
- Similarly Łukasiewicz logic $Ł_n$, for each $n \geq 3$, also has *g-MTD* and *g-MTMP* $\Delta = \{(\xi_1^{n-1} \Rightarrow \xi_2)\}$ where $(\xi_1^{n-1} \Rightarrow \xi_2)$ is $(\xi_1 \Rightarrow (\xi_1^{n-2} \Rightarrow \xi_2))$. ▽

Recall from Subsection 2.3.1 the definition of the fibring $H = \langle C, R_g, R_\ell \rangle$ of two Hilbert calculi H' and H'' (see Remark 2.3.5). Observe that the consequence system induced by H is not the union (in the sense of [280]) of the consequence systems induced by H' and H'' neither for local nor for global derivation. Moreover taking $\Gamma' \subseteq L(C', \Xi)$ and $\Gamma'' \subseteq L(C'', \Xi)$ in general we obtain that $(\Gamma')^{\vdash'^{d}_{H',\Xi}} \subsetneq (\Gamma')^{\vdash^{d}_{H,\Xi}}$ and $(\Gamma'')^{\vdash''^{d}_{H'',\Xi}} \subsetneq (\Gamma'')^{\vdash^{d}_{H,\Xi}}$. Usually in the fibred Hilbert calculus we have a much richer notion of derivation.

Remark 2.3.35 Herein, we say that a Hilbert calculus morphism $h : H \to H'$ is a signature morphism $h : C \to C'$ such that $h(R_g) \subseteq {R_g}'$ and $h(R_\ell) \subseteq {R_\ell}'$.

A signature morphism $h : C \to C'$ can be extended to

$$\overline{h} : L(C, \Xi) \to sL(C', \Xi)$$

as expected:

- $\overline{h}(c) = h(c)$, $c \in C_0$;
- $\overline{h}(\xi) = \xi$;
- $\overline{h}(c)(\varphi_1, \ldots, \varphi_k) = h(c)(\overline{h}(\varphi_1), \ldots, \overline{h}(\varphi_k))$.

We will denote $\overline{h}(\varphi)$ by $h(\varphi)$. ▽

Example 2.3.36 We consider the fibring **S4** and **K4** Hilbert calculi. Let H^0 be a propositional Hilbert calculus defined as follows:

- $C_0^0 = \{\mathbf{t}, \mathbf{f}\}$;
  $C_1^0 = \{\neg\}$;
  $C_2^0 = \{\Rightarrow\}$;
  $C_k^0 = \emptyset$ for every $k \geq 3$;

- ${R_\ell}^0$ consists of the following rules:
  $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
  $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_2 \Rightarrow \xi_3))) \rangle$;
  $\langle \emptyset, (((\neg\, \xi_1) \Rightarrow (\neg\, \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
  $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$;

Let H' be a **S4** modal Hilbert calculus and H'' a **K4** modal Hilbert calculus such that:

- $C'_0 = C''_0 = C^0_0$;
  $C'_1 = C^0_1 \cup \{\Box'\}$ and $C''_1 = C^0_1 \cup \{\Box''\}$;
  $C'_2 = C''_2 = C^0_2$;
  $C'_k = C''_k = \emptyset$ for $k \geq 3$;
- ${R_\ell}'$ is ${R_\ell}^0$ plus
  $\langle \emptyset, ((\Box'(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box'\xi_1) \Rightarrow (\Box'\xi_2))) \rangle$;
  $\langle \emptyset, ((\Box'\xi_1) \Rightarrow \xi_1) \rangle$;
  $\langle \emptyset, ((\Box'\xi_1) \Rightarrow (\Box'(\Box'\xi_1))) \rangle$;
- ${R_\ell}''$ is ${R_\ell}^0$ plus
  $\langle \emptyset, ((\Box''(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box''\xi_1) \Rightarrow (\Box''\xi_2))) \rangle$;
  $\langle \emptyset, ((\Box''\xi_1) \Rightarrow (\Box''(\Box''\xi_1))) \rangle$;
- ${R_g}' = {R_\ell}' \cup \{\langle \xi_1, (\Box'\xi_1) \rangle\}$;
- ${R_g}'' = {R_\ell}'' \cup \{\langle \xi_1, (\Box''\xi_1) \rangle\}$.

Then the constrained fibring of H' and H'' sharing H^0 is the Hilbert calculus $\langle C, R_g, R_\ell, \rangle$ with

- $C_0 = \{\mathbf{t}, \mathbf{f}\}$;
  $C_1 = \{\neg, \Box', \Box''\}$;
  $C_2 = \{\Rightarrow\}$;
  $C_k = \emptyset$ for $k \geq 3$;
- $R_g = {R_g}' \cup {R_g}''$;
- $R_\ell = {R_\ell}' \cup {R_\ell}''$.

Hence H is a bimodal logic with two unary modal operators: a necessitation operator $\Box'$ as in **S4** and a necessitation operator $\Box''$ as in **K4**. Thus, H has two necessitations and two K axioms. The morphisms involved are in this case inclusions. ∇

In the sequel, given a formula $\varphi \in L(C, \Xi)$, $\mathrm{Var}(\varphi)$ denotes the set of all schema variables occurring in φ. Similarly with respect to a set of formulas $\Gamma \subseteq L(C, \Xi)$.

Definition 2.3.37 A Hilbert calculus has the *d-Craig interpolation property* (*d-CIP*) with respect to Ξ if:

> if $\Gamma \vdash^d_{H,\Xi} \varphi$ and $\mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi) \neq \emptyset$
> then there is $\Gamma' \subseteq L(C, \mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi))$ such that $\Gamma \vdash^d_{H,\Xi} \Gamma'$ and $\Gamma' \vdash^d_{H,\Xi} \varphi$

for every $\Gamma \subseteq L(C, \Xi)$ and $\varphi \in L(C, \Xi)$. The set Γ' is said to be a *Craig interpolant* for $\Gamma \vdash^d_{H,\Xi} \varphi$. ▽

In the definition above $\Gamma \vdash^d_{H,\Xi} \Gamma'$ stands for $\Gamma \vdash^d_{H,\Xi} \varphi$ for every $\varphi \in \Gamma'$.

Again Craig interpolation can be stated in terms of finite sets.

Proposition 2.3.38 *A Hilbert calculus H has d-Craig interpolation if and only if for every $\Psi \cup \{\eta\} \subseteq L(C, \Xi)$ with Ψ finite and Var$(\Psi) \cap$ Var$(\eta) \neq \emptyset$, there is a finite Craig interpolant whenever $\Psi \vdash^d_{H,\Xi} \eta$.*

Proof. The proof from left to right is easy.

For the other implication, assume that for every $\Psi \cup \{\eta\} \subseteq L(C, \Xi)$ with Ψ finite and $\mathrm{Var}(\Psi) \cap \mathrm{Var}(\eta) \neq \emptyset$, there is a finite Craig interpolant whenever $\Psi \vdash^d_{H,\Xi} \eta$. Furthermore, assume that $\Gamma \vdash^d_{H,\Xi} \varphi$ and $\mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi) \neq \emptyset$. Then, since $\vdash^d_{H,\Xi}$ is finitary, there is a finite $\Gamma' \subseteq \Gamma$ such that $\Gamma' \vdash^d_{H,\Xi} \varphi$. We consider two cases.

(i) $\mathrm{Var}(\Gamma') \cap \mathrm{Var}(\varphi) \neq \emptyset$.

Then, by hypothesis, there is a finite Craig interpolant Φ for $\Gamma' \vdash^d_{H,\Xi} \varphi$. Moreover, Φ is also a d-Craig interpolant for $\Gamma \vdash^d_{H,\Xi} \varphi$.

(ii) $\mathrm{Var}(\Gamma') \cap \mathrm{Var}(\varphi) = \emptyset$.

Let $\gamma \in \Gamma$ be such that there is $\xi \in \mathrm{Var}(\gamma) \cap \mathrm{Var}(\varphi)$. Then $\Gamma', \gamma \vdash^d_{H,\Xi} \varphi$ with $\Gamma' \cup \{\gamma\}$ finite and $\mathrm{Var}(\Gamma' \cup \{\gamma\}) \cap \mathrm{Var}(\varphi) \neq \emptyset$ and so by hypothesis there is a finite Craig interpolant Φ for $\Gamma', \gamma \vdash^d_{H,\Xi} \varphi$. Moreover Φ is also a d-Craig interpolant for $\Gamma \vdash^d_{H,\Xi} \varphi$. ◁

The relevance of careful reasoning is measured by the fact that in some cases it is also possible to relate local and global *CIP*. That is the case of Hilbert calculi which share with modal and first-order logics the important property that we call *careful-reasoning-by-cases*. The property states that when there is a procedure which permits that hypotheses in global reasoning can be modified so as to transform a global derivation into a local derivation.

Thus, in this kind of calculi it is possible to split a global derivation in two parts: a global derivation followed by a local derivation.

Example 2.3.39 Let H be a Hilbert calculus for normal modal logic **K**. As an illustration, we observe that the following global derivation of $((\Box\xi_1) \Rightarrow (\Box\xi_2))$ from $(\xi_1 \Rightarrow \xi_2)$

1	$(\xi_1 \Rightarrow \xi_2)$	Hyp
2	$(\Box(\xi_1 \Rightarrow \xi_2))$	Nec 1
3	$((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2)))$	K
4	$((\Box\xi_1) \Rightarrow (\Box\xi_2))$	MP 2,3

in H can be transformed into a local derivation of $((\Box\xi_1) \Rightarrow (\Box\xi_2))$ from the hypothesis $(\Box(\xi_1 \Rightarrow \xi_2))$, where $(\Box(\xi_1 \Rightarrow \xi_2)) \in \{(\xi_1 \Rightarrow \xi_2)\}^{\vdash^g_{H,\Xi}}$.

A similar procedure can be used in first-order logic by means of universal closure.

∇

This motivates the following definition.

Definition 2.3.40 A Hilbert calculus H is said to have *careful-reasoning-by-cases* with respect to Ξ if for every $\Gamma \cup \{\varphi\} \subseteq L(C, \Xi)$:

if $\Gamma \vdash^g_{H,\Xi} \varphi$, then there is $\Psi \subseteq \Gamma^{\vdash^g_{H,\Xi}}$
such that $\mathrm{Var}(\Psi) \subseteq \mathrm{Var}(\Gamma)$ and $\Psi \vdash^\ell_{H,\Xi} \varphi$.

∇

The situation in a Hilbert calculus having careful-reasoning-by-cases can be visualized in Figure 2.4.

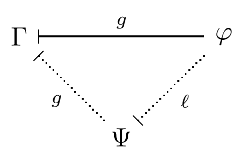


Figure 2.4: Example of careful-reasoning-by-cases

Proposition 2.3.41 *A Hilbert calculus having careful-reasoning-by-cases has global Craig interpolation whenever it has local Craig interpolation property.*

Proof. Assume that $\Gamma \vdash^g_{H,\Xi} \varphi$ and $\mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi) \neq \emptyset$. Then, since H has careful-reasoning-by-cases, there is $\Psi \subseteq L(C, \Xi)$ such that $\Gamma \vdash^g_{H,\Xi} \Psi$, $\mathrm{Var}(\Psi) \subseteq \mathrm{Var}(\Gamma)$ and $\Psi \vdash^\ell_{H,\Xi} \varphi$. There are two cases.

(i) $\mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi) \neq \emptyset$. Since H has ℓ-Craig interpolation there is a set $\Gamma' \subseteq L(C, \mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi))$ such that $\Psi \vdash^\ell_{H,\Xi} \Gamma'$ and $\Gamma' \vdash^\ell_{H,\Xi} \varphi$. Therefore, there is $\Gamma' \subseteq L(C, \mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi))$ such that

$$\Psi \vdash^g_{H,\Xi} \Gamma' \quad \text{and} \quad \Gamma' \vdash^g_{H,\Xi} \varphi$$

and so, by transitivity of $\vdash^g_{H,\Xi}$, there is $\Gamma' \subseteq L(C, \mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi))$ such that $\Gamma \vdash^g_{H,\Xi} \Gamma'$ and $\Gamma' \vdash^g_{H,\Xi} \varphi$. Since $\mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi)) \subseteq \mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi)$, then there is a set $\Gamma' \subseteq L(C, \mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi))$ such that

$$\Gamma \vdash^g_{H,\Xi} \Gamma' \ \text{ and } \ \Gamma' \vdash^g_{H,\Xi} \varphi.$$

(ii) $\mathrm{Var}(\Psi) \cap \mathrm{Var}(\varphi) = \emptyset$. Take $\gamma \in \Gamma$ such that $\mathrm{Var}(\gamma) \cap \mathrm{Var}(\varphi) \neq \emptyset$. Then $\Gamma \vdash^g_{H,\Xi} \Psi \cup \{\gamma\}$, $\mathrm{Var}(\Psi) \cup \mathrm{Var}(\gamma) \subseteq \mathrm{Var}(\Gamma)$ and $\Psi \cup \{\gamma\} \vdash^\ell_{H,\Xi} \varphi$. And we can now proceed in a similar way to case (i). ◁

Careful-reasoning-by-cases can be expressed in terms of finite sets as indicated in the following result.

Proposition 2.3.42 *A Hilbert calculus H has careful-reasoning-by-cases if and only if for every $\Psi \cup \{\eta\} \subseteq L(C, \Xi)$ with Ψ finite such that $\Psi \vdash^g_{H,\Xi} \eta$, there is a finite Ψ' such that $Var(\Psi') \subseteq Var(\Psi)$, $\Psi' \subseteq \Psi^{\vdash^g_{H,\Xi}}$ and $\Psi' \vdash^\ell_{H,\Xi} \eta$.*

Example 2.3.43 Some illustrations of Craig interpolation can be given:

- We can conclude that the modal Hilbert calculi referred to above have local Craig interpolation.
- Since modal Hilbert calculi have careful-reasoning-by-cases, then the modal Hilbert calculi referred to above have global Craig interpolation. ▽

In order to investigate the preservation of interpolation, we must be able to *transform* derivations in the fibring into derivations in the components (the other way around we already know how to do). For this purpose, we start by translating formulas from a Hilbert calculus to another in the presence of a Hilbert calculus morphism. Assume that $h : H \to H'$ is a Hilbert calculus morphism. Take

$$\Xi^\bullet = \Xi \cup \{\xi_{c'(\varphi_1,\ldots,\varphi_k)} : c'(\varphi_1, \ldots, \varphi_k) \in L(C', \Xi), c' \in C'_k \setminus h(C_k)\}$$

as a new set of variables. Take Ξ' as $\Xi^\bullet \setminus \Xi$. Each $\xi_{c'(\varphi_1,\ldots,\varphi_k)}$ is a *ghost* of $(c'(\varphi_1, \ldots, \varphi_k))$ in H and will only have an auxiliary role. The introduction of ghosts is similar to the introduction of surrogates used in [281] for proving preservation of properties in the fusion of modal logics sharing the propositional connectives.

Definition 2.3.44 Let $h : C \to C'$ be a signature morphism and let $\Xi^\bullet$ as above. The *translation*

$$\tau : L(C', \Xi) \to L(C, \Xi^\bullet)$$

is a map defined inductively as follows:

- $\tau(\xi) = \xi$ for $\xi \in \Xi$;
- $\tau(h(c)) = c$ for $c \in C_0$;

- $\tau(c') = \xi_{c'}$ for $c' \in C'_0 \setminus h(C_0)$;
- $\tau(h(c)(\gamma'_1, \ldots, \gamma'_k)) = (c(\tau(\gamma'_1), \ldots, \tau(\gamma'_k)))$ for $c \in C_k$ and $\gamma'_1, \ldots, \gamma'_k \in L(C', \Xi)$;
- $\tau(c'(\gamma'_1, \ldots, \gamma'_k)) = \xi_{c'(\gamma'_1,\ldots,\gamma'_k)}$ for $c' \in C'_k \setminus h(C_k)$ and $\gamma'_1, \ldots, \gamma'_k \in L(C', \Xi)$.

The substitution $\tau^{-1} : \Xi^\bullet \to L(C', \Xi)$ is defined as follows:

- $\tau^{-1}(\xi) = \xi$ for $\xi \in \Xi$;
- $\tau^{-1}(\xi_{c'(\gamma'_1,\ldots,\gamma'_k)}) = (c'(\gamma'_1, \ldots, \gamma'_k))$
  for $(c'(\gamma'_1, \ldots, \gamma'_k)) \in L(C', \Xi)$ and $c' \in C'_k \setminus h(C_k)$. ∇

The following are technical lemmas that will be needed to relate derivations in H' with derivations in H.

Lemma 2.3.45 *Let $h : C \to C'$ be a signature morphism, $\sigma' : \Xi \to L(C', \Xi)$ and $\sigma : \Xi^\bullet \to L(C, \Xi^\bullet)$ substitutions such that*

$$\sigma(\xi) = \tau(\sigma'(\xi))$$

for every $\xi \in \Xi$. Then $\sigma(\gamma) = \tau(\sigma'(h(\gamma)))$ for every $\gamma \in L(C, \Xi)$.

Lemma 2.3.46 *If $h : C \to C'$ is a signature morphism, then $\tau^{-1} \circ h \circ \tau = id$.*

We are now ready to relate global derivations in $h(H)$ with global derivations in H where:

$$h(H) = \langle C', h(R_g), h(R_\ell)\rangle.$$

Of course in H' we can prove more things than in $h(H)$ since in $h(H)$ no rules in ${R_g}' \setminus h(R_g)$ can be used.

Lemma 2.3.47 *Let $h : H \to H'$ be a Hilbert calculus morphism. Then, for every $\Gamma' \cup \{\psi'\} \subseteq L(C', \Xi)$,*

$$\Gamma' \vdash^g_{h(H),\Xi} \psi' \text{ if and only if } \tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\psi').$$

Proof. First, assume that $\Gamma' \vdash^g_{h(H),\Xi} \psi'$. We prove that $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\psi')$ by induction on the length n of a proof of ψ' from Γ'.

Base:

a) ψ' is obtained from an instance of the axiom $\langle\emptyset, h(\varphi)\rangle$ in $h(H)$ with substitution $\sigma' : \Xi \to L(C', \Xi)$. Then $\emptyset \vdash^g_{h(H).\Xi} \sigma'(h(\varphi))$. Let $\sigma : \Xi^\bullet \to L(C, \Xi^\bullet)$ be a substitution such that

$$\sigma(\xi) = \tau(\sigma'(\xi))$$

for every $\xi \in \Xi$. Hence $\emptyset \vdash^g_{H,\Xi^\bullet} \sigma(\varphi)$, by monotonicity $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \sigma(\varphi)$ and since $\sigma'(h(\varphi)) = \psi'$ we get $\tau(\sigma'(h(\varphi)) = \tau(\psi')$ and so using Lemma 2.3.45 $\sigma(\varphi) = \tau(\psi')$.

b) Straightforward when ψ' is an hypothesis.

Step: Assume that ψ' is an instance of $h(\varphi)$ in the proof rule $\langle\{\delta_1,\ldots,\delta_k\},\varphi\rangle$ in H with substitution $\sigma' : \Xi \to L(C', \Xi)$. Then

$$\Gamma' \vdash^g_{h(H),\Xi} \sigma'(h(\delta_i))$$

for $i = 1,\ldots,k$ and so by the induction hypothesis $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\sigma'(h(\delta_i)))$ for $i = 1,\ldots,k$. Taking substitution σ such that $\sigma(\xi) = \tau(\sigma'(\xi))$ for every $\xi \in \Xi$ then $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \sigma(\delta_i)$ for $i = 1,\ldots,k$, hence $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \sigma(\varphi)$ and so $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\psi')$.

Suppose now that $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\psi')$. Then, since Hilbert calculus morphisms preserve derivations, $h(\tau(\Gamma')) \vdash^g_{h(H),\Xi^\bullet} h(\tau(\psi'))$, so

$$\tau^{-1}(h(\tau(\Gamma'))) \vdash^g_{h(H),\Xi} \tau^{-1}(h(\tau(\psi')))$$

since derivations are closed for substitutions. Thus, using Lemma 2.3.46, we get $\Gamma' \vdash^g_{h(H),\Xi} \psi'$. ◁

A similar result can be stated for local derivations. Derivations in H with different sets of variables can also be related.

Lemma 2.3.48 *Let $h : H \to H'$ be a Hilbert calculus morphism. Assume that Γ is finite,*

$$\delta_1 \ldots \delta_n$$

is a derivation of $\Gamma \vdash^d_{H,\Xi^\bullet} \varphi$ and Υ' is the set of variables in Ξ' occurring in the derivation. Let Υ be a set of variables in Ξ not occurring in the derivation such that $|\Upsilon'| = |\Upsilon|$ and μ a bijection from Υ' to Υ. Consider a substitution $\sigma : \Xi^\bullet \to L(C, \Xi^\bullet)$ such that:

$$\begin{cases} \sigma(\xi) = \xi & \text{for } \xi \in \Xi; \\ \sigma(\upsilon) = \mu(\upsilon) & \text{for } \upsilon \in \Upsilon'. \end{cases}$$

Then $\sigma(\Gamma) \vdash^d_{H,\Xi} \sigma(\varphi')$.

Proof. The sequence $\sigma(\delta_1)\ldots\sigma(\delta_n)$ is a derivation of $\sigma(\varphi')$ from $\sigma(\Gamma')$ using variables in Ξ. ◁

Lemma 2.3.49 *Let $h : H \to H'$ be a Hilbert calculus morphism. Assume that Γ is finite,*

$$\delta_1,\ldots,\delta_n$$

is a derivation of $\Gamma \vdash^d_{H,\Xi} \varphi$ and take a subset Υ of the set of variables in Ξ occurring in the derivation. Let Υ' be a set of variables in Ξ' such that $|\Upsilon'| = |\Upsilon|$ and μ a bijection from Υ to Υ'. Consider a substitution $\sigma : \Xi \to L(C, \Xi^\bullet)$ such that:

$$\begin{cases} \sigma(\xi) = \xi & \text{for } \xi \in \Xi; \\ \sigma(\upsilon) = \mu(\upsilon) & \text{for } \upsilon \in \Upsilon. \end{cases}$$

Then $\sigma(\Gamma) \vdash^d_{H',\Xi^\bullet} \sigma(\varphi)$.

Proof. The sequence $\sigma(\delta_1)\dots\sigma(\delta_n)$ is a derivation of $\sigma(\varphi)$ from $\sigma(\Gamma)$ using variables in $\Xi^\bullet$. $\triangleleft$

Lemma 2.3.47 states the relationship between derivations in H (over $\Xi \cup \Xi'$) with (parts of) derivations in H' (over Ξ) using only rules in H. Lemmas 2.3.48 and 2.3.49 are needed for getting preservation of derivations from H' using variables in Ξ to H using variables from $\Xi \cup \Xi'$ and vice-versa.

We are ready to investigate preservation of different kinds of interpolation by fibring. In the presence of fibring we have to deal with the ghost variables for each component Hilbert calculus as well as two translations. We start by investigating preservation of careful-reasoning-by-cases.

Now we prove a technical lemma about preservation of careful-reasoning-by-cases when changing the set of variables. This is the situation which occurs when a morphism $h : H \to H'$ is present and we want to transfer derivations from H' to H.

Lemma 2.3.50 *Let H be a Hilbert calculus with careful-reasoning-by-cases with respect to Ξ and $h : H \to H'$ a Hilbert calculus morphism. Then H also has careful-reasoning-by-cases with respect to $\Xi^\bullet$.*

Proof. Assume that $\Gamma \vdash^g_{H,\Xi^\bullet} \varphi$ where Γ is finite. Pick up a derivation of φ from Γ. Let Υ' be the set of variables in Ξ' appearing in the derivation, μ a bijection to a set Υ of variables in Ξ not occurring in that derivation and σ an assignment as required in Lemma 2.3.48. Then, by the same lemma,

$$\sigma(\Gamma) \vdash^g_{H,\Xi} \sigma(\varphi).$$

Since H has careful-reasoning-by-cases with respect to Ξ, using Proposition 2.3.42, there is a finite set $\Phi \subseteq L(C, \Xi)$ such that:

(1) $\mathrm{Var}(\Phi) \subseteq \mathrm{Var}(\sigma(\Gamma))$;

(2) $\Phi \subseteq \sigma(\Gamma)^{\vdash^g_{H,\Xi}}$;

(3) $\Phi \vdash^\ell_{H,\Xi} \sigma(\varphi)$.

Consider substitution σ^{-1} defined from bijection μ^{-1} as in Lemma 2.3.48. Then, there is a finite set $\sigma^{-1}(\Phi) \subseteq L(C, \Xi^\bullet)$ such that:

(i) $\mathrm{Var}(\sigma^{-1}(\Phi)) \subseteq \mathrm{Var}(\Gamma)$;

(ii) $\sigma^{-1}(\Phi) \subseteq \Gamma^{\vdash^g_{H,\Xi^\bullet}}$;

(iii) $\sigma^{-1}(\Phi) \vdash^\ell_{H,\Xi^\bullet} \varphi$.

In fact, (i) and (iii) follow from (1) and (3) above, respectively. In order to prove (ii), assume that $\psi \in \Phi$. Then $\sigma(\Gamma) \vdash^g_{H,\Xi} \psi$ by (2) above, and so, by Lemma 2.3.49, $\sigma^{-1}(\sigma(\Gamma)) \vdash^g_{H,\Xi^\bullet} \sigma^{-1}(\psi)$. Thus, $\Gamma \vdash^g_{H,\Xi^\bullet} \sigma^{-1}(\psi)$.

From (i), (ii) and (iii) follows that H has careful-reasoning-by-cases with respect to $\Xi^\bullet$. ◁

Before analyzing the preservation of careful-reasoning-by-cases by a morphism we need two more lemmas.

Lemma 2.3.51 *Let $h : H \to H'$ be a Hilbert calculus morphism and $\Psi \vdash^d_{H,\Xi^\bullet} \varphi$ where Ψ is finite. Let $\Upsilon \subseteq \Xi'$ be the set of variables used in a derivation of φ from Ψ. Choose μ and σ as in Lemma 2.3.48 and take also μ^{-1} and σ^{-1} defined from bijection μ^{-1} as in the same Lemma. Then for every $\gamma \in L(C, \Upsilon)$, we have*

$$\sigma^{-1}(h(\sigma(\gamma))) = h(\gamma).$$

Proof. The proof follows by induction on the structure of γ. We only consider as an illustration the case of $\gamma \in \Upsilon$:

$$\sigma^{-1}(h(\sigma(\gamma))) = \sigma^{-1}(h(\mu(\gamma))) = \sigma^{-1}(\mu(\gamma)) = \mu^{-1}(\mu(\gamma)) = \gamma = h(\gamma).$$

◁

We can now relate derivation in H and $h(H)$ over the set of variables $\Xi^\bullet$.

Lemma 2.3.52 *Let $h : H \to H'$ be a Hilbert calculus morphism. Then*

$$h(\Gamma) \vdash^d_{h(H),\Xi^\bullet} h(\varphi) \text{ whenever } \Gamma \vdash^d_{H,\Xi^\bullet} \varphi.$$

Proof. Assume that $\Gamma \vdash^d_{H,\Xi^\bullet} \varphi$. Then by Lemma 2.3.48, choosing μ and σ as indicated there,

$$\sigma(\Gamma) \vdash^d_{H,\Xi} \sigma(\varphi)$$

and, since morphisms preserve derivations over variables in Ξ,

$$h(\sigma(\Gamma)) \vdash^d_{h(H),\Xi} h(\sigma(\varphi)).$$

Using Lemma 2.3.49, choosing μ^{-1} and σ^{-1},

$$\sigma^{-1}(h(\sigma(\Gamma))) \vdash^d_{h(H),\Xi^\bullet} \sigma^{-1}(h(\sigma(\varphi)))$$

and so, by Lemma 2.3.51, $h(\Gamma) \vdash^d_{h(H),\Xi^\bullet} h(\varphi)$. ◁

Proposition 2.3.53 *If H is a Hilbert calculus having careful-reasoning-by-cases with respect to Ξ and $h : H \to H'$ is a Hilbert calculus morphism, then $h(H)$ also has careful-reasoning-by-cases with respect to Ξ.*

Proof. Assume that H has careful-reasoning-by-cases with respect to Ξ and $\Gamma' \vdash^g_{h(H),\Xi} \psi'$. Then $\tau(\Gamma') \vdash^g_{H,\Xi^\bullet} \tau(\psi')$ by Proposition 2.3.47. Since, by Lemma 2.3.50, H has careful-reasoning-by-cases with respect to $\Xi^\bullet$, there is set $\Phi \subseteq L(C, \Xi^\bullet)$ finite and such that:

- $\text{Var}(\Phi) \subseteq \text{Var}(\tau(\Gamma'))$;
- $\Phi \subseteq \tau(\Gamma')^{\vdash^g_{H,\Xi^\bullet}}$;
- $\Phi \vdash^\ell_{H,\Xi^\bullet} \tau(\psi')$.

Hence there is a finite set $h(\Phi) \subseteq L(C', \Xi^\bullet)$ such that:

- $\text{Var}(h(\Phi)) \subseteq \text{Var}(h(\tau(\Gamma')))$;
- $h(\Phi) \subseteq h(\tau(\Gamma'))^{\vdash^g_{h(H),\Xi^\bullet}}$ using Lemma 2.3.52;
- $h(\Phi) \vdash^\ell_{h(H),\Xi^\bullet} h(\tau(\psi'))$ using Lemma 2.3.52.

Moreover, considering the substitution τ^{-1} introduced in Definition 2.3.44, there is a finite set $\tau^{-1}(h(\Phi)) \subseteq L(C', \Xi)$ such that:

- $\text{Var}(\tau^{-1}(h(\Phi))) \subseteq \text{Var}(\tau^{-1}(h(\tau(\Gamma'))))$;
- $\tau^{-1}(h(\Phi)) \subseteq \tau^{-1}(h(\tau(\Gamma')))^{\vdash^g_{h(H),\Xi}}$, using closure for substitution;
- $\tau^{-1}(h(\Phi)) \vdash^\ell_{h(H),\Xi} \tau^{-1}(h(\tau(\psi')))$, using closure for substitution.

Hence, by Lemma 2.3.46, there is a finite set $\tau^{-1}(h(\Phi)) \subseteq L(C', \Xi)$ such that:

- $\text{Var}(\tau^{-1}(h(\Phi))) \subseteq \text{Var}(\Gamma')$;
- $\tau^{-1}(h(\Phi)) \subseteq \Gamma'^{\vdash^g_{h(H),\Xi}}$;
- $\tau^{-1}(h(\Phi)) \vdash^\ell_{h(H),\Xi} \psi'$.

Therefore, $h(H)$ has careful-reasoning-by-cases with respect to Ξ. $\triangleleft$

We want to investigate the preservation of careful-reasoning-by-cases by the fibring. We start by setting-up the ghost variables and the translations. Let H be the fibring of H' and H''. Then we need to work with:

- $\Xi'' = \{\xi_{c''(\varphi_1,\dots,\varphi_k)} : i''(c'')(\varphi_1, \dots, \varphi_k) \in L(C, \Xi), c'' \in C''_k\}$;
- $\Xi' = \{\xi_{c'(\varphi_1,\dots,\varphi_k)} : i'(c')(\varphi_1, \dots, \varphi_k) \in L(C, \Xi), c' \in C'_k\}$;

as the ghosts of H'' in H' and of H' in H'', respectively. Let

- $\tau' : L(C, \Xi) \to L(C', \Xi \cup \Xi'')$;
- $\tau'' : L(C, \Xi) \to L(C'', \Xi \cup \Xi')$;

be the translations and τ'^{-1} and τ''^{-1} substitutions as defined in Definition 2.3.44. Recall that $i'(H') = \langle C, i'({R_g}'), i'({R_\ell}'), \rangle$ and $i''(H'') = \langle C, i''({R_g}''), i''({R_\ell}''), \rangle$ are Hilbert calculi with the same connectives as the fibring but where only the rules from H' and H'' can be used, respectively.

Theorem 2.3.54 *Careful-reasoning-by-cases is preserved by fibring Hilbert calculi.*

Proof. Let H be the fibring of two Hilbert calculi H' and H'' both having careful-reasoning-by-cases with respect to Ξ. Assume that Γ is finite and that $\Gamma \vdash^g_{H,\Xi} \varphi$. Suppose, additionally, that there exists a global derivation of φ from Γ in H of the form

$$\underbrace{\varphi_1 \ldots \varphi_k}_{i''(H'')\text{ rules}} \qquad \underbrace{\varphi_{k+1} \ldots \varphi_n}_{i'(H')\text{ rules}}$$

such that $\varphi_1 \ldots \varphi_k$, $\varphi_{k+1} \ldots \varphi_n$ were justified by rules in $i''(H'')$ and $i'(H')$, respectively and $\Gamma_1 \subseteq \Gamma$ is the part of Γ used in the derivation until step k and $\Gamma_2 \subseteq \Gamma$ is the part of Γ used from $k+1$ onwards. Then

$$\Gamma_1 \vdash^g_{i''(H''),\Xi} \varphi_k.$$

Assume also that, as a simplification, only φ_k is used as premise of a rule applied in $\varphi_{k+1} \ldots \varphi_n$. Since H'' has careful-reasoning-by-cases with respect to Ξ then, by Proposition 2.3.53, $i''(H'')$ has careful-reasoning-by-cases with respect to Ξ and so there is a finite set $\Phi \subseteq L(C, \Xi)$ such that:

- $\mathrm{Var}(\Phi) \subseteq \mathrm{Var}(\Gamma_1)$;
- $\Phi \subseteq {\Gamma_1}^{\vdash^g_{H,\Xi}}$;
- $\Phi \vdash^\ell_{H,\Xi} \varphi_k$.

Hence

$$\Gamma_2, \Phi \vdash^g_{H,\Xi} \varphi.$$

Since H' has careful-reasoning-by-cases with respect to Ξ then, by Proposition 2.3.53, $i'(H')$ has careful-reasoning-by-cases with respect to Ξ and so there is a finite set $\Psi \subseteq L(C, \Xi)$ such that:

- $\mathrm{Var}(\Psi) \subseteq \mathrm{Var}(\Gamma_2) \cup \mathrm{Var}(\Phi)$;
- $\Psi \subseteq (\Gamma_2 \cup \Phi)^{\vdash^g_{H,\Xi}}$;
- $\Psi \vdash^\ell_{H,\Xi} \varphi$.

Therefore, there is a finite set $\Psi \subseteq L(C, \Xi)$ such that:

- $\mathrm{Var}(\Psi) \subseteq \mathrm{Var}(\Gamma)$;

- $\Psi \subseteq \Gamma^{\vdash^g_{H,\Xi}}$;
- $\Psi \vdash^\ell_{H,\Xi} \varphi$.

Hence H has careful-reasoning-by-cases with respect to Ξ. The case where in $\varphi_{k+1} \dots \varphi_n$ more than one element of $\varphi_1 \dots \varphi_k$ is used is proved in a similar way. The same applies to the case where more than two blocks of rules from H'' and H' are applied. ◁

Example 2.3.55 We provide an illustration of careful-reasoning-by-cases in the context of modal logics. Let H be the fibring of two modal Hilbert calculi H' and H'' that share the propositional part, in particular $\Rightarrow$ and $\neg$ but have two different modalities $\Box'$ and $\Box''$ (recall Example 2.3.36).

Consider the following global derivation of

$$\{(\Box''(\varphi' \Rightarrow \varphi'')), (\Box'(\Box''\varphi'))\} \vdash^g_{H,\Xi} ((\Box'(\Box''\varphi'')) \vee \psi')$$

(note the use of the necessitation rule Nec$'$) using variables in Ξ:

1	$(\Box''(\varphi' \Rightarrow \varphi''))$	Hyp
2	$((\Box''(\varphi' \Rightarrow \varphi'')) \Rightarrow ((\Box''\varphi') \Rightarrow (\Box''\varphi'')))$	K$''$
3	$((\Box''\varphi') \Rightarrow (\Box''\varphi''))$	MP 1,2
4	$(\Box'((\Box''\varphi') \Rightarrow (\Box''\varphi'')))$	Nec$'$ 3
5	$((\Box'((\Box''\varphi') \Rightarrow (\Box''\varphi''))) \Rightarrow ((\Box'(\Box''\varphi')) \Rightarrow (\Box'(\Box''\varphi''))))$	K$'$
6	$((\Box'(\Box''\varphi')) \Rightarrow (\Box'(\Box''\varphi'')))$	MP 4,5
7	$(\Box'(\Box''\varphi'))$	Hyp
8	$(\Box'(\Box''\varphi''))$	MP 6,7
9	$((\Box'(\Box''\varphi'')) \vee \psi')$	$\vee$I 8

Observe that steps 1 and 2 are justified by rules in H'' and that all the other steps are justified by rules in H' (since the implication is shared step 3 can be seen as an hypothesis in H'). In step 9, we assume the usual definition of disjunction in terms of $\Rightarrow$ and $\neg$.

Hence, from the derivation above, we extract a global derivation in H' of $((\Box'\xi_{(\Box''\varphi'')}) \vee \psi')$ from the set of hypotheses $\{(\xi_{(\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')}), (\Box'\xi_{(\Box''\varphi')})\}$, using variables in Ξ but also ghost variables in Ξ'':

1	$(\xi_{(\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')})$	Hyp
2	$(\Box'(\xi_{(\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')}))$	Nec$'$ 1
3	$((\Box'(\xi_{((\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')})) \Rightarrow ((\Box'\xi_{(\Box''\varphi')}) \Rightarrow (\Box'\xi_{(\Box''\varphi'')})))$	K$'$
4	$((\Box'\xi_{(\Box''\varphi')}) \Rightarrow (\Box'\xi_{(\Box''\varphi'')}))$	MP 2,3
5	$(\Box'\xi_{(\Box''\varphi')})$	Hyp
6	$(\Box'\xi_{(\Box''\varphi'')})$	MP 5,4
7	$((\Box'\xi_{(\Box''\varphi'')}) \vee \psi')$	$\vee$I 6

Using the fact that H' has careful-reasoning-by-cases with respect to Ξ, by Lemma 2.3.50, it also has careful-reasoning-by-cases with respect to $\Xi^\bullet$, and taking

$$\Psi' = \{(\Box'(\xi_{(\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')})), (\Box'\xi_{(\Box''\varphi')})\}$$

we have

$$\Psi' \vdash^{\ell}_{H'} ((\Box' \xi_{(\Box''\varphi'')}) \vee \psi') \text{ and } \Psi' \subseteq \{(\xi_{(\Box''\varphi')} \Rightarrow \xi_{(\Box''\varphi'')}), (\Box' \xi_{(\Box''\varphi')})\}^{\vdash^{g}_{H'}}.$$

Hence $\{(\Box'((\Box''\varphi') \Rightarrow (\Box''\varphi''))), (\Box'(\Box''\varphi'))\}$ is such that:

- $\{(\Box'((\Box''\varphi') \Rightarrow (\Box''\varphi''))), (\Box'(\Box''\varphi'))\} \vdash^{\ell}_{H,\Xi} ((\Box'(\Box''\varphi'')) \vee \psi')$;
- $\{(\Box''(\varphi' \Rightarrow \varphi'')), (\Box'(\Box''\varphi'))\} \vdash^{g}_{H,\Xi} (\Box'((\Box''\varphi') \Rightarrow (\Box''\varphi'')))$.

This example illustrates that careful-reasoning-by-cases still holds in the bimodal logic with the modalities $\Box'$ and $\Box''$. ∇

The objective now is to show that Craig interpolation is preserved by fibring under mild conditions. By using the concept of bridge we are able to provide preservation of Craig interpolation in general, that is, for both global and local reasoning.

Definition 2.3.56 A *d-bridge* to the Hilbert calculus H' in the fibring H of Hilbert calculi H' and H'' sharing H^0 is a pair $\langle h_1, h_2 \rangle$ of maps from $L(C, \Xi)$ to $L(C', \Xi)$ such that

- $h_1(\Psi) \vdash^{d}_{i'(H'),\Xi} h_2(\varphi)$ whenever $\Psi \vdash^{d}_{H,\Xi} \varphi$
- $\text{var}(h_i(\varphi)) = \text{var}(\varphi)$
- $\gamma \vdash^{d}_{H,\Xi} h_1(\gamma)$ and $h_2(\delta) \vdash^{d}_{H,\Xi} \delta$ for any γ and δ in $L(C, \Xi)$

where i' is the morphism from H' to H, $\Psi \subseteq L(C, \Xi)$ and $\varphi \in L(C, \Xi)$.

We now present an example of a bridge involving intuitionistic and classical propositional Hilbert calculi.

Example 2.3.57 Consider the fibring H of the Hilbert calculus H_c for classical logic where

- C_{c0} contains $\bot_c$ and a denumerable set of propositional symbols
- $C_{c2} = \{\to_c\}$

and the Hilbert calculus H_i for intuitionistic logic where

- C_{i0} contains $\bot_i$ and a denumerable set of variables containing the classical propositional symbols
- $C_{i2} = \{\wedge_i, \vee_i, \to_i\}$

sharing a Hilbert calculus H^0 where ${C^0}_0$ is the set of classical propositional symbols and the other components are empty. Consider the map h_1 from $L(C, \Xi)$ to $L(C_i, \Xi)$ inductively defined as follows

- $h_1(\varphi) = \varphi$ whenever φ is in Ξ
- $h_1(\perp_c) = \perp_i$
- $h_1(\varphi_1 \to_c \varphi_2) = (\neg_i\neg_i h_1(\varphi_1)) \to_i (\neg_i\neg_i h_1(\varphi_2))$
- $h_1(c_i(\varphi_1, \ldots, \varphi_n)) = c_i(\neg_i\neg_i h_1(\varphi_1), \ldots, \neg_i\neg_i h_1(\varphi_n))$ if c_i is in C_{in}, $n \geq 0$.

Then the pair

$$\langle h_1, h_2 \rangle$$

where h_2 is $\lambda\varphi.\neg_i\neg_i h_1(\varphi)$ constitute a bridge to H_i in H since considering h_c a map from $L(C, \Xi)$ to $L(C, \Xi)$ inductively defined as follows

- $h_c(\varphi) = \varphi$ whenever φ is either $\perp_i$ or $\perp_c$
- $h_c(\varphi) = \neg_c\neg_c\varphi$ whenever φ is in $\Xi \cup C_{i0}$ and is neither $\perp_i$ nor $\perp_c$
- $h_c(\varphi_1 \to_c \varphi_2) = (\neg_c\neg_c h_c(\varphi_1)) \to_c (\neg_c\neg_c h_c(\varphi_2))$
- $h_c(c_i(\varphi_1, \ldots, \varphi_n)) = c_i(\neg_c\neg_c h_c(\varphi_1), \ldots, \neg_c\neg_c h_c(\varphi_n))$ if c_i is in C_{in} and $n > 0$.

It happens that

1. $h_1(\Psi) \vdash^d_{i_i(H_i),\Xi} h_2(\varphi)$ whenever $\Psi \vdash^d_{H,\Xi} \varphi$
2. $h_2(\varphi) \dashv\vdash^d_{H,\Xi} h_c(\varphi)$
3. $h_1(\varphi) \dashv\vdash^d_{H,\Xi} h_c(\varphi)$
4. $h_c(\varphi) \dashv\vdash^d_{H,\Xi} \varphi$. ∇

We now show that Craig interpolation can be preserved by constrained or unconstrained fibring whenever there is a bridge in the fibring.

Theorem 2.3.58 *d-Craig interpolation holds in the Hilbert calculus resulting from constrained or unconstrained fibring provided that one of the component Hilbert calculi has d-Craig interpolation and there is a d-bridge to that Hilbert calculus in the fibring.*

Proof. Let H be the fibring of Hilbert calculi H' and H'' sharing H^0. Note that H can be the unconstrained fibring of H' and H''. Assume without loss of generality that H' has d-Craig interpolation and that there is a d-bridge $\langle h_1, h_2 \rangle$ to H' in H. Let $\Gamma \subseteq L(C, \Xi)$ be finite, $\varphi \in L(C, \Xi)$ and assume that $\Gamma \vdash^d_{H,\Xi} \varphi$ and that $\mathrm{Var}(\Gamma) \cap \mathrm{Var}(\varphi) \neq \emptyset$. Then

$$h_1(\Gamma) \vdash^d_{i'(H'),\Xi} h_2(\varphi)$$

and $\text{var}(h_1(\Gamma)) \cap \text{var}(h_2(\varphi)) \neq \emptyset$ since h_1 and h_2 constitute a d-bridge, see Definition 2.3.56. Taking into account Lemma 2.3.47 then

$$\tau'(h_1(\Gamma)) \vdash^d_{H',\Xi\cup\Xi''} \tau'(h_2(\varphi))$$

and since H' has d-Craig interpolation with respect to $\Xi\cup\Xi''$ by Proposition 2.3.38, there is a finite set $\Psi' \subseteq L(C', \Xi \cup \Xi'')$ such that

- $\text{Var}(\Psi') \subseteq \text{Var}(\tau'(\Gamma)) \cap \text{Var}(\tau'(\varphi))$
- $\tau'(h_1(\Gamma)) \vdash^d_{H',\Xi\cup\Xi''} \Psi'$;
- $\Psi' \vdash^d_{H',\Xi\cup\Xi''} \tau'(h_2(\varphi))$;

therefore

- $\text{Var}(i'(\Psi')) \subseteq \text{Var}(i'(\tau'(\Gamma))) \cap \text{Var}(i'(\tau'(\varphi)))$;
- $i'(\tau'(h_1(\Gamma))) \vdash^d_{H,\Xi\cup\Xi''} i'(\Psi')$;
- $i'(\Psi') \vdash^d_{H,\Xi\cup\Xi''} i'(\tau'(h_2(\varphi)))$;

and so, by Lemma 2.3.46,

- $\text{Var}(\tau'^{-1}(i'(\Psi'))) \subseteq \text{Var}(\Gamma) \cap \text{Var}(\varphi)$;
- $h_1(\Gamma) \vdash^d_{H,\Xi} \tau'^{-1}(i'(\Psi'))$
- $\tau'^{-1}(i'(\Psi')) \vdash^d_{H,\Xi} h_2(\varphi)$.

Henceforth $\tau'^{-1}(i'(\Psi'))$ is a d-Craig interpolant for $\Gamma \vdash^d_{H,\Xi} \varphi$ since $\Gamma \vdash^d_{H,\Xi} h_1(\Gamma)$ and $h_2(\varphi) \vdash^d_{H,\Xi} \varphi$. ◁

Note that in the proof of Theorem 2.3.58 it is not required the preservation of the metatheorems of modus ponens and deduction.

Example 2.3.59 The fibring of the propositional intuitionistic Hilbert calculus and the propositional classical Hilbert calculus sharing only the classical propositional symbols has local Craig interpolation. Indeed Theorem 2.3.58 can be applied, see Example 2.3.57. ▽

Interpolation in the presence of deductive implication is discussed in the next theorem. We also need the concept of deductive conjunction.

Definition 2.3.60 A Hilbert calculus H has (binary) *deductive conjunction* with respect to global derivation if there is $\wedge \in C_2$ such that

- $(\xi_1 \wedge \xi_2) \vdash^g_{H,\Xi} \xi_i$ for $i = 1, 2$;

- $\{\xi_1, \xi_2\} \vdash^g_{H,\Xi} (\xi_1 \wedge \xi_2)$. ▽

Theorem 2.3.61 *d-Theoremhood-Craig interpolation holds in the Hilbert calculus resulting from constrained or unconstrained fibring of Hilbert calculi H' and H'' provided that:*

- *H' has d-deductive implication;*
- *H' has d-Craig interpolation;*
- *H'' has d-deductive conjunction.*
- *there is a d-bridge to H' in the fibring.*

Proof. Let H be the fibring of Hilbert calculi H' and H'' sharing H^0 and that one of the component Hilbert calculi has d-Craig interpolation and there is a d-bridge to that Hilbert calculus in the fibring.

Note that H can be the unconstrained fibring of H' and H''. Assume without loss of generality that H' has d-deductive implication and that H'' has d-deductive conjunction. Suppose that $\vdash^d_{H,\Xi} (\varphi_1 \Rightarrow' \varphi_2)$ and $\mathrm{Var}(\varphi_1) \cap \mathrm{Var}(\varphi_2) \neq \emptyset$. Then by d-MTMP, $\varphi_1 \vdash^d_{H,\Xi} \varphi_2$ and $\mathrm{Var}(\varphi_1) \cap \mathrm{Var}(\varphi_2) \neq \emptyset$ and so by Theorem 2.3.58 there is a finite set $\Psi \subseteq L(C, \Xi)$ such that

$$\mathrm{Var}(\Psi) \subseteq \mathrm{Var}(\varphi_1) \cap \mathrm{Var}(\varphi_2)$$

and

$$\varphi_1 \vdash^d_{H,\Xi} \Psi \quad \text{and} \quad \Psi \vdash^d_{H,\Xi} \varphi_2$$

The result follows by taking

$$({\textstyle\bigwedge}''_{\psi\in\Psi} \psi)$$

as an interpolant and using d-MTD. ◁

A similar result can be obtained when changing the roles of H' and H''.

Example 2.3.62 The fibring of the propositional intuitionistic Hilbert calculus and the propositional classical Hilbert calculus sharing only classical propositional symbols has local theoremhood-Craig interpolation. ▽

The technique that was used to proving the preservation of interpolation can be summarized in Figure 2.5. Let:

- Hil be the class of Hilbert calculi with global Craig interpolation;
- Hil^+ be the subclass of Hil composed by Hilbert calculi that have deductive conjunction, deductive implication and a bridge.

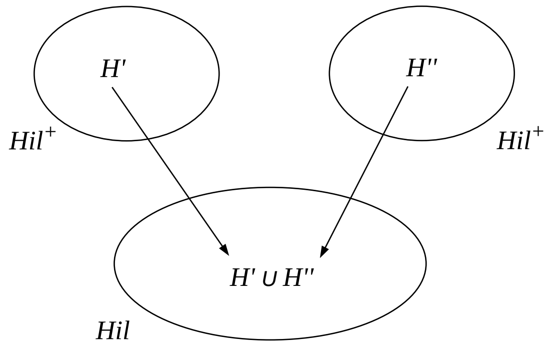


Figure 2.5: Preservation of interpolation

If $H', H'' \in Hil^+$ then the unconstrained fibring $H' \cup H'' \in Hil$ using Theorem 2.3.58.

Craig interpolation is constructive in the deductive system resulting from the unconstrained or constrained fibring of deductive systems where there is a bridge to one of the components having also that property whenever 1) Craig interpolation is constructive in that deductive system, and 2) it is constructive the procedure of obtaining a deduction in the deductive system with Craig interpolation for each deduction in the fibring.

We analyze the time complexity of the algorithm I to obtain the interpolant of derivations in the fibring, described in the proof of Theorem 2.3.58, assuming that 1) there is an algorithm I° to obtain the interpolant in the component deductive system with Craig interpolation and 2) there is an algorithm $I_{\langle h_1,h_2\rangle}$ that given a deduction in the fibring and the bridge $\langle h_1, h_2\rangle$ returns a similar deduction in the deductive system enjoying Craig interpolation. In order to obtain a time complexity result it is important to consider the size (in bits) of the derivation. For derivation $\varphi_1 \ldots \varphi_k$, the size is

$$||\varphi_1 \ldots \varphi_k|| = \sum_{i=1,\ldots,k} ||\varphi_i||$$

where $||\varphi_i||$ is the number of bits required to represent (efficiently) the formula φ_i for $i = 1, \ldots, n$. We denote by $Time(I^\circ, D)$ the cost in time of applying algorithm I° to deduction D, and similarly for the algorithms I and $I_{\langle h_1,h_2\rangle}$.

Proposition 2.3.63 *Let $\mathcal{D}$ be the fibring of deductive systems such that there is a bridge $\langle h_1, h_2\rangle$ to a component deductive system $\mathcal{D}^\circ$ with Craig interpolation. Assume that*

- $Time(I^\circ, D^\circ) \in O(f^\circ(||D^\circ||))$ *for each derivation D° in $\mathcal{D}^\circ$*

- $Time(I_{\langle h_1,h_2\rangle}, D) \in O(f_{\langle h_1,h_2\rangle}(||D||))$ *for each derivation* D *in* $\mathcal{D}$

then

$$Time(I, D) \in O(f_{\langle h_1,h_2\rangle}(||D||) + f^\circ(||I_{\langle h_1,h_2\rangle}(D)||))$$

for each derivation D *of* $\Gamma \vdash_{\mathcal{D},\Xi} \varphi$ *in* $\mathcal{D}$.

We omit the proof of the proposition since it follows straightforwardly. Observe that if I° and $I_{\langle h_1,h_2\rangle}$ take polynomial time so does I. Of course if, for instance, I° takes exponential time and $I_{\langle h_1,h_2\rangle}$ takes polynomial time then I also takes, in the worst case, exponential time.

2.4 Final remarks

In this chapter we investigated the concept of fibring in a homogeneous setting where both component logics are presented as Hilbert calculi. The need for defining fibring starting by detailing signatures was discussed. In particular, we stressed that the formulas of the fibring should involve symbols from both component logics. It was also emphasized the importance of presenting the calculi with schema variables as an essential ingredient for being able to use rules from both logics applied to mixed formulas.

It should be stressed that fibring of Hilbert calculi can be seen as a universal construction in the sense that the derivations obtained from the components should be derivations in the fibring (preservation of derivations) and, moreover, that fibring is the "minimal" Hilbert calculus with that property. In this introductory chapter we avoided the use of categorial concepts as much as possible. We preferred to present the basic notions in set-theoretic terms.

The chapter could be written choosing another kind of deductive system like for example sequent calculi, natural deduction calculi and tableau systems. For an investigation of fibring in the context of natural deduction systems see [226] and of tableau systems see [76, 17]. In most cases labeled systems are considered, using possible worlds as labels (see [16, 270]). We decided to use Hilbert calculi because they are simpler, in the sense that they do not require provisos (like for example fresh variables in natural deduction even for propositional based logics). In Chapter 4 we will address the issue of fibring logics presented through sequent calculus and tableau calculus, without using possible worlds as labels.

We believe that it is worthwhile to investigate fibring of, say, sequent calculus labeled with truth-values, taking into account some results in [203]. Also of interest would be to obtain preservation results for cut elimination following the ideas presented in [200].

The chapter dedicated some effort to illustrate preservation properties. Among them, we can mention the preservation of metatheorems like the metatheorem of the modus ponens, the metatheorem of deduction and the metatheorem of congruence. These metatheorems are relevant for proving the preservation of completeness in Chapter 3.

Hilbert calculi having careful reasoning (meaning that they support local and global reasoning) were also discussed. Even for stating metatheorems the distinction between local and global reasoning is of utmost importance.

The chapter also dealt with Hilbert calculi that have the property that global reasoning can be converted into local reasoning, called careful reasoning by cases. This issue is of relevance for the preservation of local interpolation. The chapter ends with a tour on interpolation. Preservation of global and local interpolation is discussed. Interpolation has not any follow up in the next chapters but the so called "ghost" technique is also used in Chapter 4 when dealing with heterogeneous fibring.

Preservation of decidability is worthwhile to be investigated in this context.

Chapter 3

Splicing logics: Semantic fibring

In this chapter we will concentrate on the semantic aspects of logics and their fibring. Things may not be so easy as in the syntactic case discussed in Chapter 2, since logics may have (usual) semantics from quite distinct nature. For instance, knowledge is characterized by a normal modal logic, while obligation is often characterized by a non-normal modal logic (see [54], Chapter 11). To achieve our goal we adopt the algebraic viewpoint as a semantic framework. This is justified taking into account that general algebras augmented with suitable operators to represent connectives and modalities constitute a wide abstraction, in the tradition initiated in [159].

Herein, we have a first glimpse of heterogeneous combination. That is, we may have original logics with different semantic structures. However, the semantic fibring mechanism assumes that the logics to be combined are presented by ordered algebras. Hence, as a first step we have to say how the original semantic structures induce ordered algebras. Therefore, throughout this chapter, when given a particular logic we have to say how its usual semantic structures induce algebraic structures in this sense.

An additional step consists on putting together Hilbert calculi and interpretation systems (defined as collections of algebraic structures) thus obtaining what we call a logic system. In this context, we can therefore consider soundness and completeness properties.

An essential issue in combination of logics in general, and fibring in particular, is the investigation of preservation of properties. Some steps in that direction were given in Chapter 2, where several results on preservation of metatheorems and interpolation were established. Herein, we study preservation of soundness and completeness properties by fibring. In many cases, we are only able to give sufficient conditions for preservation.

In Section 3.1, we start by introducing algebraic interpretation structures and interpretation systems. Then we define fibring of interpretation systems. We illustrate the concepts with several examples including classical logic, modal logics (**K**, **S4** and **B**), intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 3.2, we introduce the notions of logic system, soundness and completeness. In Section 3.3, we discuss the preservation of soundness and completeness properties. In Section 3.4, we establish the relationship between the present approach and the fibring by functions. We discuss how the algebraic semantic structures represent the point-based semantic approach to fibring: to a model and a point in the fibring corresponds a model in each component logics, which means that to a model of the fibring we can associate several models for each component logic. We do so by showing that we can establish a one to one relationship between a model of the fibring and a model in each component by assuming that the interpretation systems are closed for unions. In Section 3.5 we present some final remarks.

This chapter capitalizes on the work developed in [237, 240, 282]. The interested reader can also have a look at [37] and [41] in [10].

3.1 Interpretation systems

As noted above, dealing with semantic aspects of logics is also an important issue in fibring, in particular, when we are interested in preservation of properties like soundness and completeness. Semantics of fibring is more difficult since the homogeneous scenario is clearly not common. Indeed logics tend to be presented by different semantic structures. Therefore, the first goal is to find the adequate semantic for the components of the fibring retaining the properties of the original semantics. This semantic unit is an algebra. So we have an additional task consisting of showing that the algebraic semantics keeps the properties of the original semantics.

An *algebra* is a tuple

$$\mathcal{B} = \langle B, \{f_i\}_{i\in I}\rangle$$

where B is a non-empty set (the carrier set), I a non-empty set and each f_i is a map (operation) $f_i : B^{k_i} \to B$ for some $k_i \in \mathbb{N}$. When $k_i = 0$, f_i is said to be a *constant*. Giving a signature C, an algebra $\mathcal{B}$ over C consists of a carrier set B and a *denotation map* $\nu_k(c) : B^k \to B$ for each $c \in C_k$ and $k \in \mathbb{N}$. We represent such algebras as a pair

$$\langle B, \nu\rangle$$

where $\nu = \{\nu_k\}_{k\in\mathbb{N}}$ is the indexed family of maps $\nu_k : C_k \to B^{(B^k)}$ (as usual, X^Y denotes the set of all maps from the set Y to the set X). Among the algebras over the signature C, we have the so called free algebra. A *free algebra* $\mathcal{B}$ over C is an algebra over C whose carrier B is $L(C)$ and $\nu_k(c)(\varphi_1, \ldots, \varphi_k) = c(\varphi_1, \ldots, \varphi_k)$. A *free algebra* over C *generated* by a set A is the free algebra over C' where $C'_0 = C_0 \cup A$ and $C'_k = C_k$ for $k > 0$.

In order to ensure the preservation of some properties by fibring, it is convenient to consider enriched algebras, called interpretation structures. The underlying idea is that the elements of the carrier sets of those algebras are truth values. In such case, each algebra is endowed with a partial order (ordered algebra) which reflects derivability. Moreover, these partial orders are required to have a top element.

Definition 3.1.1 An *interpretation structure* over the signature C is a tuple

$$\mathcal{B} = \langle B, \leq, \nu, \top \rangle$$

where $\langle B, \leq, \top \rangle$ is a partial order with top $\top$ and $\langle B, \nu \rangle$ is an algebra over C. $\triangledown$

The set B is the set of *truth values* and $\top$ is the *designated value* whose intended purpose is to state when a formula is true in a structure. The relation $\leq$ allows the comparison between truth values. In this chapter, we are not considering logics where the set of designated values is not a singleton (as in the case of some finitely many-valued logics). In Chapter 9, the reader can find details about combining logics where there are more than one designated value.

Example 3.1.2 Recall the classical signature C presented in Example 2.1.4. Let $\langle B, \sqcap, \sqcup, -, \top, \bot \rangle$ be a Boolean algebra [232], that is, an algebra where the operations $\sqcap, \sqcup : B^2 \to B$, $- : B \to B$ and the constants $\top$, $\bot$ are such that, for every $b, b_1, b_2 \in B$,

- $\sqcap$ and $\sqcup$ are commutative and associative
- $(b_1 \sqcap b_2) \sqcup b_2 = b_2$ and $(b_1 \sqcup b_2) \sqcap b_2 = b_2$;
- $\sqcap$ is distributive with respect to $\sqcup$ and $\sqcup$ is distributive with respect to $\sqcap$;
- $-(-b) = b$;
- $-(b_1 \sqcup b_2) = (-b_1) \sqcap (-b_2)$ and $-(b_1 \sqcap b_2) = (-b_1) \sqcup (-b_2)$;
- $b \sqcap \top = b$ and $\bot \sqcap b = \bot$;
- $b \sqcup (-b) = \top$ and $b \sqcap (-b) = \bot$.

The interpretation structure over C induced by a Boolean algebra is $\langle B, \leq, \nu, \top \rangle$ where, for every $b, b_1, b_2 \in B$,

- $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
- $\nu_1(\neg)(b) = -b$;
- $\nu_2(\Rightarrow)(b_1, b_2) = (-b_1) \sqcup b_2$.

An interesting particular case of Boolean algebra, to be used along this book, is the two-elements Boolean algebra

$$\mathit{2} = \langle 2, \sqcap, \sqcup, -, 1, 0 \rangle$$

where 2 is the set $\{0, 1\}$. The interpretation structure over C induced by $\mathit{2}$ as above will be denoted by $\mathcal{B}_{\mathit{2}}$. $\triangledown$

Example 3.1.3 We recast the example above, but now considering the classical signature C presented in Example 2.1.2. That is, the unique difference with the example above is that now $C_0 = \mathbb{P}$, the set of propositional symbols. Thus, in the context of Boolean algebras, the propositional symbols are to be interpreted, as usual, as arbitrary elements in the Boolean algebra of the given structure by means of a valuation map $V : \mathbb{P} \to B$. More specifically, let $\langle B, \sqcap, \sqcup, -, \top, \bot \rangle$ be a Boolean algebra. The interpretation structure over C induced by the given Boolean algebra and a valuation $V : \mathbb{P} \to B$ is $\langle B, \leq, \nu, \top \rangle$, where $\leq$ and ν are as in Example 3.1.2, with the following additional clause for ν:

- $\nu_0(p) = V(p)$ for $p \in \mathbb{P}$.

In the particular case of the two–elements Boolean algebra $\mathcal{2}$, the interpretation structure over C induced by $\mathcal{2}$ and a valuation mapping $V : \mathbb{P} \to B$ will be denoted by $\mathcal{B}_{\mathcal{2}}^{V}$. ∇

This approach to the definition of interpretation structures induced by the original semantics will be followed below. Other examples are the following.

Example 3.1.4 Recall the intuitionistic signature C introduced in Example 2.1.4. Let $\langle B, \sqcap, \sqcup, \to, \top, \bot \rangle$ be a Heyting algebra [232], that is, an algebra where the operations $\sqcap, \sqcup : B^2 \to B$ and $\to : B^2 \to B$ and the constants $\top$, $\bot$ are such that, for every $b, b_1, b_2 \in B$,

- $\sqcap$ and $\sqcup$ are commutative and associative
- $(b_1 \sqcap b_2) \sqcup b_2 = b_2$ and $(b_1 \sqcup b_2) \sqcap b_2 = b_2$;
- $b_1 \to (b_2 \sqcap b_3) = (b_1 \to b_2) \sqcap (b_1 \to b_3)$;
- $b_1 \sqcap (b_1 \to b_2) = b_1 \sqcap b_2$ and $(b_1 \to b_2) \sqcap b_2 = b_2$;
- $(b_1 \to b_1) \sqcap b_2 = b_2$;
- $b \sqcap \top = b$ and $\bot \sqcap b = \bot$.

The interpretation structure over C induced by a Heyting algebra is $\langle B, \leq, \nu, \top \rangle$ where:

- $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
- $\nu_1(\neg)(b) = b \to \bot$;
- $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
- $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
- $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \to b_2$.

In every interpretation structure induced by a Heyting algebra,

$$b_1 \leq (b_2 \to b_3) \text{ if and only if } (b_1 \sqcap b_2) \leq b_3.$$

Since every Boolean algebra is also an Heyting algebra, this result also holds in interpretation structures induced by Boolean algebras. ▽

Example 3.1.5 As it was done with classical logic, the example above can be extended to cope with the intuitionistic signature introduced in Example 2.1.2, that is, by setting $C_0 = \mathbb{P}$. Thus, the interpretation structure over C induced by a Heyting algebra $\langle B, \sqcap, \sqcup, \to, \top, \bot\rangle$ and a valuation mapping $V : \mathbb{P} \to B$ is $\langle B, \leq, \nu, \top\rangle$, where $\leq$ and ν are defined as in Example 3.1.4, with the following addendum:

- $\nu_0(p) = V(p)$ for $p \in \mathbb{P}$. ▽

Example 3.1.6 Recall the modal signature C presented in Example 2.1.4. Let $\langle W, R\rangle$ be a Kripke frame, that is, W is a non-empty set (the set of worlds) and $R \subseteq W \times W$ (the accessibility relation). The interpretation structure over C induced by $\langle W, R\rangle$ is $\langle B, \leq, \nu, \top\rangle$ where, for every $b, b_1, b_2 \in B$,

- B is $\wp W$;
- $b_1 \leq b_2$ if and only if $b_1 \cap b_2 = b_1$;
- $\nu_1(\neg)(b) = W \setminus b$;
- $\nu_1(\Box)(b) = \{w \in W : \text{ if } wRw' \text{ then } w' \in b\}$;
- $\nu_2(\Rightarrow)(b_1, b_2) = (W \setminus b_1) \cup b_2$;
- $\top$ is W. ▽

Example 3.1.7 Consider now the modal signature C presented in Example 2.1.2. We proceed as in the example above, but now considering Kripke structure instead of Kripke frames. More precisely, let $\langle W, R, V\rangle$ be a Kripke structure, that is, $\langle W, R\rangle$ is a Kripke frame and $V : \mathbb{P} \to \wp W$ is a valuation mapping. The interpretation structure over C induced by $\langle W, R, V\rangle$ is defined as in Example 3.1.6, but now including the following clause:

- $\nu_0(p) = V(p)$ for $p \in \mathbb{P}$. ▽

Example 3.1.8 Consider again the modal signature C presented in Example 2.1.4. Let $\langle B, \sqcap, \sqcup, -, \top, \bot, N\rangle$ be a modal algebra (introduced by McColl, see [228] and [134]), that is, $\langle B, \sqcap, \sqcup, -, \top, \bot\rangle$ is a Boolean algebra and $N : B \to B$ is an operation such that, for every $b_1, b_2 \in B$,

- $N(\top) = \top$;
- $N(b_1 \sqcap b_2) = N(b_1) \sqcap N(b_2)$.

The interpretation structure induced by a modal algebra is $\langle B, \leq, \nu, \top \rangle$ where, for every $b, b_1, b_2 \in B$,

- $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
- $\nu_1(\neg)(b) = -b$;
- $\nu_1(\Box)(b) = N(b)$;
- $\nu_2(\Rightarrow)(b_1, b_2) = (-b_1) \sqcup b_2$.

If we consider the modal signature introduced in Example 2.1.2, we must add a valuation map $V : \mathbb{P} \to B$ and extend the mapping ν as expected: $\nu_0(p) = V(p)$ for every $p \in \mathbb{P}$. ∇

Other interesting examples are related to many-valued logics. Many-valued logics are a rich source of examples in non-classical logics; the first examples were proposed by the Polish logician Jan Łukasiewicz in 1920 (see [184] and [27] for an English version), with the aim of using a third truth-value to represent "possibility" in order to overcome some philosophical difficulties posed by Aristotle in his well-known questions about the future contingent.

Example 3.1.9 Recall that the signature for Łukasiewicz logics is the same as the one for classical logic. A 3-valued Łukasiewicz algebra [137] is an algebra $\langle B, \oplus, \ominus, \bot \rangle$ where B has three elements and $\oplus : B^2 \to B$, $\ominus : B \to B$ are operations such that, for every $b, b_1, b_2 \in B$,

- $\oplus$ is commutative and associative;
- $\ominus(\ominus b) = b$;
- $b \oplus \bot = b$ and $b \oplus (\ominus \bot) = \ominus \bot$;
- $(\ominus((\ominus b_1) \oplus b_2)) \oplus b_2 = (\ominus((\ominus b_2) \oplus b_1)) \oplus b_1$.

$\top$ abbreviates $\ominus\bot$ and the operations $\otimes$, $\sqcap$, $\sqcup$ and $\sqsupset$ are also defined as abbreviations:

- $b_1 \otimes b_2 = \ominus(\ominus b_1 \oplus \ominus b_2)$;
- $b_1 \sqsupset b_2 = (\ominus b_1) \oplus b_2$;
- $b_1 \sqcup b_2 = (b_1 \otimes (\ominus b_2)) \oplus b_2$;
- $b_1 \sqcap b_2 = (b_1 \oplus (\ominus b_2)) \otimes b_2$.

The typical 3-valued Łukasiewicz algebra is as follows:

- $B = \{\perp, 1/2, \top\}$;
- $\ominus 1/2 = 1/2$, $\ominus\top = \perp$ and $\ominus\perp = \top$;
- $1/2 \oplus 1/2 = \top$, $b \oplus \top = \top$ and $b \oplus \perp = b$.

Any other 3-valued Łukasiewicz algebra is isomorphic to this one.

Suppose that we choose the classical signature introduced in Example 2.1.4. The interpretation structure induced by a 3-valued Łukasiewicz algebra is $\langle B, \leq, \nu, \top\rangle$ where, for every $b, b_1, b_2 \in B$,

- $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
- $\nu_1(\neg)(b) = \ominus b$;
- $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \sqsupset b_2$.

The connectives $\wedge$ and $\vee$ can be defined as abbreviations. Alternatively, its denotations are $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$ and $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$.

On the other hand, if we choose the classical signature introduced in Example 2.1.2, we define the interpretation structure induced by a 3-valued Łukasiewicz algebra and a valuation $V : \mathbb{P} \to B$ as above, by stipulating that $\nu_0(p) = V(p)$ for every $p \in \mathbb{P}$.

The reader should observe that 3-valued Łukasiewicz logic is sound and complete with respect to the above algebraic semantics. ∇

In an attempt to understand intuitionistic logic in terms of several truth-values, Kurt Gödel in 1932 (see [101]) proposed what now are known as the Gödel logics. Since it was also proved in [101] that intuitionistic logic cannot be characterized by finitely many-valued logics, Gödel logics stay as a kind of approximation to intuitionistic logic.

Example 3.1.10 Recall that the signature for Gödel logics is the same as the one for intuitionistic logic. A 3-valued Gödel algebra is an algebra $\langle B, \sqcap, \sqcup, \sqsupset, \ominus, \top, \perp\rangle$ where B has three elements, $\sqcap$, $\sqcup$, $\perp$, $\top$ are as in a Heyting algebra and $\sqsupset$: $B^2 \to B$, $\ominus : B \to B$ are operations such that, for every $b, b_1, b_2 \in B$,

- $b_1 \sqsupset b_2 = \begin{cases} \top & \text{if } b_1 \sqcap b_2 = b_1 \\ b_2 & \text{otherwise;} \end{cases}$
- $\ominus b = \begin{cases} \top & \text{if } b = \perp \\ \perp & \text{otherwise.} \end{cases}$

The typical 3-valued Gödel algebra has $B = \{\perp, 1/2, \top\}$ and $b_1 \sqsupset b_2 = \top$ if $b_1 = b_2 = 1/2$, $b_1 = \perp$ or $b_2 = \top$ and $b_1 \sqsupset b_2 = b_2$ otherwise. Other 3-valued Gödel algebras are isomorphic to this one.

Starting from the intuitionistic signature introduced in Example 2.1.4, the interpretation structure induced by a 3-valued Gödel algebra is $\langle B, \leq, \nu, \top \rangle$ where, for every $b, b_1, b_2 \in B$,

- $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
- $\nu_1(\neg)(b) = \ominus b$;
- $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
- $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
- $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \sqsupset b_2$.

If we now consider the intuitionistic signature introduced in Example 2.1.2, the interpretation structure induced by a 3-valued Gödel algebra and a valuation map $V : \mathbb{P} \to B$ is defined as above, extending the definition of ν as expected: $\nu_0(p) = V(p)$ for every $p \in \mathbb{P}$.

As in the previous case, it is also to be observed that 3-valued Gödel logic is indeed sound and complete with respect to the above algebraic semantics. ∇

Semantics give a meaning to formulas: the idea is that a formula will denote a truth-value in a particular interpretation structure. Observing the Definition 3.1.1, it is clear that the schema variables do not have a meaning within an interpretation structure. For this purpose we need to introduce assignments.

Definition 3.1.11 An *assignment* over an interpretation structure $\mathcal{B}$ is a map $\alpha : \Xi \to B$.

Assignments play the role of the valuation maps with respect to schema variables. In fact, propositional symbols in $\mathbb{P}$ are interpreted by means of valuations (see the examples above), in the same manner as schema variables are interpreted by assignments. This lead us to the following definition.

Definition 3.1.12 Let $\mathcal{B} = \langle B, \leq, \nu, \top \rangle$ be an interpretation structure and α an assignment over $\mathcal{B}$. The *denotation* map $[\![\cdot]\!]^\alpha_\mathcal{B} : L(C) \to B$ is inductively defined as follows:

- $[\![\xi]\!]^\alpha_\mathcal{B} = \alpha(\xi)$;
- $[\![c]\!]^\alpha_\mathcal{B} = \nu_0(c)$ for every $c \in C_0$;
- $[\![c(\varphi_1, \ldots, \varphi_k)]\!]^\alpha_\mathcal{B} = \nu_k(c)([\![\varphi_1]\!]^\alpha_\mathcal{B}, \ldots, [\![\varphi_k]\!]^\alpha_\mathcal{B})$ whenever $c \in C_k$, $k \geq 1$ and $\varphi_1, \ldots, \varphi_k \in L(C)$. ∇

Remark 3.1.13 Note that, using an assignment together with an interpretation structure, all the formulas can be interpreted, even the ones without propositional symbols in $\mathbb{P}$. In particular, consider all the interpretation structures in the examples above constructed over the signatures described in Example 2.1.4. Then, these interpretation structures can now interpret every formula of the corresponding signatures with the help of assignments. The assignments are therefore used to interpret the atomic formulas.

The semantics of formulas given above is truth-functional, by the very definition. In other words, the denotation of a k-ary connective c, in a given structure, is a k-ary function. Moreover, the truth-value of a formula built up from k components using connective c depends functionally on the truth-values of these components (using that function). From a technical point of view, the denotation mapping $[\![\cdot]\!]^{\alpha}_{\mathcal{B}} : L(C) \to B$ is an homomorphism of algebras over C. An homomorphism, between two algebras over C, $h : \mathcal{B} \to \mathcal{B}'$ is a map $h : B \to B'$ such that:

- $h(\nu_0(c)) = \nu'_0(c)$;
- $h(\nu_k(c)(b_1, \ldots, b_k)) = \nu'_k(c)(h(b_1), \ldots, h(b_k))$.

However, there are several logics that do not admit a truth-functional treatment as, for instance, the paraconsistent logic $\mathfrak{C}_1$ (see Example 2.2.9). In Chapter 5 we propose a suitable semantic approach for non-truth-functional logics and their fibring. ∇

Now we give some examples of denotations of formulas.

Example 3.1.14 Recall Example 3.1.4. For every $n \in \mathbb{N}$, let $n^{\geq} = \{i \in \mathbb{N} : n \leq i\}$ and note that $0^{\geq} = \mathbb{N}$. Consider the Heyting algebra $\langle B, \cap, \cup, \to, \mathbb{N}, \emptyset \rangle$ where $B = \{\emptyset\} \cup \{n^{\geq} : n \in \mathbb{N}\}$ and $b_1 \to b_2 = \mathbb{N}$ if $b_1 \subseteq b_2$ and $b_1 \to b_2 = b_2$ otherwise. Let $\mathcal{B} = \langle B, \nu, \leq, \mathbb{N} \rangle$ be the interpretation structure induced by the Heyting algebra above, and consider the assignment $\alpha : \Xi \to B$ such that $\alpha(\xi_1) = 4^{\geq}$. Then

- $[\![\neg \xi_1]\!]^{\alpha}_{\mathcal{B}} = \emptyset$;
- $[\![(\neg(\neg \xi_1)) \Rightarrow \xi_1]\!]^{\alpha}_{\mathcal{B}} = 4^{\geq}$.

On the other hand, for every assignment α over $\mathcal{B}$,

- $[\![\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)]\!]^{\alpha}_{\mathcal{B}} = \mathbb{N}$;
- $[\![\xi_1 \Rightarrow (\neg(\neg \xi_1))]\!]^{\alpha}_{\mathcal{B}} = \mathbb{N}$. ∇

Example 3.1.15 Recall Example 3.1.6. The Kripke frame $\langle W, R \rangle$, where

- $W = \{w_1, w_2\}$;
- $w_1 R w_2$ and $w_2 R w_2$,

induces the interpretation structure $\mathcal{B} = \langle B, \nu, \leq, W\rangle$ where

$$B = \{\emptyset, \{w_1\}, \{w_2\}, W\}.$$

Consider the assignment $\alpha : \Xi \to B$ such that $\alpha(\xi_1) = \{w_1\}$. Then

- $[\![\Box\xi_1]\!]^\alpha_\mathcal{B} = \emptyset$;
- $[\![\xi_1 \Rightarrow (\Box\xi_1)]\!]^\alpha_\mathcal{B} = \{w_2\}$.

On the other hand, for every interpretation structure $\mathcal{B}$ induced by a Kripke frame $\langle W, R\rangle$ and every assignment α over $\mathcal{B}$,

- $[\![(\neg\xi_1) \Rightarrow (\xi_1 \Rightarrow \xi_2)]\!]^\alpha_\mathcal{B} = W$;
- $[\![(\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2))]\!]^\alpha_\mathcal{B} = W$. ▽

Several interpretation structures can be defined for each signature C. Interpretation systems include a signature plus a class of interpretation structures for that signature. In most of the cases not all interpretation structures are relevant.

Definition 3.1.16 An *interpretation system* is a pair

$$I = \langle C, \mathcal{A}\rangle$$

where C is a signature and $\mathcal{A}$ is a class of interpretation structures over C. ▽

We give examples of interpretation systems for several logics which will be useful for providing illustrations for the fibring.

Example 3.1.17 An interpretation system for classical propositional logic **CPL** can be defined as follows:

$$I_{\mathbf{CPL}} = \langle C, \mathcal{A}_{\mathbf{CPL}}\rangle$$

where C is the classical signature presented in Example 2.1.4 and $\mathcal{A}_{\mathbf{CPL}}$ is the class of all the interpretation structures $\mathcal{B}$ over C induced by a Boolean algebra (recall Example 3.1.2).

By a well-known result (see, for instance, [227]), it is enough to consider the Boolean algebra $\mathit{2}$. Thus, another interpretation system for **CPL** is

$$I_{\mathit{2}} = \langle C, \{\mathcal{B}_{\mathit{2}}\}\rangle$$

where $\mathcal{B}_{\mathit{2}}$ is the structure induced by the Boolean algebra $\mathit{2}$. The assignments play the role of the valuation maps. ▽

Example 3.1.18 It is possible to recast the example above but now including the set $\mathbb{P}$ in the signature. Thus, define the interpretation system

$$\widehat{I}_{\mathbf{CPL}} = \langle C, \widehat{\mathcal{A}}_{\mathbf{CPL}} \rangle$$

for classical propositional logic, where C is the classical signature presented in Example 2.1.2, and $\widehat{\mathcal{A}}_{\mathbf{CPL}}$ is the class of all the interpretation structures $\mathcal{B}$ over C induced by a Boolean algebra and a valuation (recall Example 3.1.3).

In particular, let $\mathcal{B}_{\mathit{2}}^V$ be the structure induced by the Boolean algebra $\mathit{2}$ and a valuation $V : \mathbb{P} \to 2$. Then we obtain another interpretation system for **CPL**:

$$\widehat{I}_{\mathit{2}} = \langle C, \{\mathcal{B}_{\mathit{2}}^V \ : \ V \in 2^{\mathbb{P}}\} \rangle.$$

$\triangledown$

Example 3.1.19 An interpretation system $I = \langle C, \mathcal{A} \rangle$ for intuitionistic logic can be defined as follows: C is the intuitionistic signature presented in Example 2.1.4, and $\mathcal{A}$ is the class of all the interpretation structures $\mathcal{B}$ over C induced by a Heyting algebra (recall Example 3.1.4).

Another possibility is to consider $\widehat{I} = \langle \widehat{C}, \widehat{\mathcal{A}} \rangle$, where $\widehat{C}$ is the intuitionistic signature presented in Example 2.1.2, and $\widehat{\mathcal{A}}$ is the class of all the interpretation structures $\mathcal{B}$ over $\widehat{C}$ induced by a Heyting algebra and a valuation (recall Example 3.1.5). $\triangledown$

Example 3.1.20 An interpretation system $I_{\mathbf{S4}}$ for modal logic **S4** includes the modal signature presented in Example 2.1.4 and the class $\mathcal{A}_{\mathbf{S4}}$ of all the interpretation structures induced by Kripke frames whose accessibility relation is reflexive and transitive (recall Example 3.1.6).

An interpretation system $I_{\mathbf{B}}$ for modal logic **B** is obtained by considering the modal signature as above and the class $\mathcal{A}_{\mathbf{B}}$ of the all interpretation structures induced by Kripke frames whose accessibility relation is symmetric.

An interpretation system $I_{\mathbf{K}}$ for modal logic **K** is obtained as above but considering now the class $\mathcal{A}_{\mathbf{K}}$ of all interpretation structures induced by Kripke frames.

If we consider now the modal signature presented in Example 2.1.2, the corresponding interpretation systems $\widehat{I}_{\mathbf{S4}}$, $\widehat{I}_{\mathbf{B}}$ and $\widehat{I}_{\mathbf{K}}$ are obtained, but now using Kripke structures (recall Example 3.1.7).

We get another interpretation system for modal logic **K** when we consider the class of all interpretation structures induced by a modal algebra, as described in Example 3.1.8. $\triangledown$

Example 3.1.21 A (3-valued) Gödel interpretation system over the corresponding signature introduced in Example 2.1.4 is obtained by taking the class of all interpretation structures induced by 3-valued Gödel algebras, as described in Example 3.1.10. A (3-valued) Łukasiewicz interpretation system over the corresponding signature introduced in Example 2.1.4 is obtained by considering the class of all interpretation structures induced by 3-valued Łukasiewicz algebra, as described in Example 3.1.9.

If we include the set $\mathbb{P}$ of propositional symbols, we must consider the interpretation structures induced by the corresponding algebras plus a valuation mapping. ∇

Given an interpretation system, it is important to know when a formula is a semantic consequence of a set of formulas.

Definition 3.1.22 Let $I = \langle C, \mathcal{A} \rangle$ be an interpretation system. We say that a formula $\varphi \in L(C)$ is *entailed* by a set of formulas $\Gamma \subseteq L(C)$ within I, written

$$\Gamma \vDash_I \varphi$$

if, for every interpretation structure $\mathcal{B}$ in $\mathcal{A}$ and assignment α over $\mathcal{B}$, if $[\![\gamma]\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma \in \Gamma$ then $[\![\varphi]\!]_{\mathcal{B}}^{\alpha} = \top$. ∇

When $\emptyset \vDash_I \varphi$ we say that φ is *valid* in I and just write $\vDash_I \varphi$. We denote by $\Gamma^{\vDash_I}$ the set of formulas that are entailed by Γ in I.

An interpretation system $I = \langle C, \mathcal{A} \rangle$ induces the closure operator $\vDash_I$ and the consequence system $\mathcal{C}(I) = \langle C, \vDash_I \rangle$. Before presenting the proof of this result, we present an auxiliary lemma.

Lemma 3.1.23 *Let C be a signature and $\mathcal{B}$ an interpretation structure over C. For every $\varphi \in L(C)$, every assignment α over $\mathcal{B}$ and every substitution σ on C, we have*

$$[\![\sigma(\varphi)]\!]_{\mathcal{B}}^{\alpha} = [\![\varphi]\!]_{\mathcal{B}}^{\alpha'}$$

where α' is the assignment over $\mathcal{B}$ such that $\alpha'(\xi) = [\![\sigma(\xi)]\!]_{\mathcal{B}}^{\alpha}$ for each $\xi \in \Xi$.

Proof. The result is easily established using induction. $\triangleleft$

Proposition 3.1.24 *An interpretation system $I = \langle C, \mathcal{A} \rangle$ induces a structural consequence system $\mathcal{C}(I) = \langle C, \vDash_I \rangle$.*

Proof. Let $\Gamma, \Gamma_1, \Gamma_2 \subseteq L(C)$.

Extensiveness: Clearly $\Gamma \vDash_I \gamma$ for every $\gamma \in \Gamma$ and therefore $\Gamma \subseteq \Gamma^{\vDash_I}$.

Monotonicity: Assume $\Gamma_1 \subseteq \Gamma_2$ and $\Gamma_1 \vDash_I \varphi$. Given an interpretation structure $\mathcal{B}$ in $\mathcal{A}$ and an assignment α over $\mathcal{B}$ such that $[\![\gamma]\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma \in \Gamma_2$, then also $[\![\gamma]\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma \in \Gamma_1$ and therefore $[\![\varphi]\!]_{\mathcal{B}}^{\alpha} = \top$. Hence, $\Gamma_2 \vDash_I \varphi$.

Idempotence: Assume $\Gamma^{\vDash_I} \vDash_I \varphi$. Given an interpretation structure $\mathcal{B}$ in $\mathcal{A}$ and an assignment α over $\mathcal{B}$ such that $[\![\gamma]\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma \in \Gamma$, then $[\![\gamma']\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma' \in \Gamma^{\vDash_I}$ and therefore $[\![\varphi]\!]_{\mathcal{B}}^{\alpha} = \top$. Hence, $\Gamma \vDash_I \varphi$.

Structurality: Assume $\Gamma \vDash_I \varphi$. Let $\mathcal{B}$ be an interpretation in $\mathcal{A}$ and α an assignment over $\mathcal{B}$ such that $[\![\sigma(\gamma)]\!]_{\mathcal{B}}^{\alpha} = \top$ for each $\gamma \in \Gamma$. Thus, using Lemma 3.1.23, $[\![\gamma]\!]_{\mathcal{B}}^{\alpha'} = \top$ for each $\gamma \in \Gamma$, where α' is the assignment on $\mathcal{B}$ such that $\sigma'(\xi) =$

$[\![\sigma(\xi)]\!]^{\alpha}_{\mathcal{B}}$ for each $\xi \in \Xi$. Since $\Gamma \vDash_I \varphi$, $[\![\varphi]\!]^{\alpha'}_{\mathcal{B}} = \top$ and, using again Lemma 3.1.23, $[\![\sigma(\varphi)]\!]^{\alpha}_{\mathcal{B}}$. Hence, $\sigma(\Gamma) \vDash_I \sigma(\varphi)$. ◁

Our purpose is to define fibring of interpretation systems. Before that, we need to introduce the notion of reduct of an interpretation structure.

Definition 3.1.25 Let C, C' be signatures such that $C \leq C'$ and consider the interpretation structure $\mathcal{B}' = \langle B', \leq', \nu', \top' \rangle$ over C'. The *reduct* of $\mathcal{B}'$ to C is the interpretation structure

$$\mathcal{B}'|_C = \langle B', \leq', \nu'_{|C}, \top' \rangle$$

over C such that $\nu'_{|C k}(c) = \nu'_k(c)$ for every $c \in C_k$. ▽

The reduct has the same truth-values as $\mathcal{B}'$ and the denotations of constructors of C in the reduct are the same as in $\mathcal{B}'$. Hence, a reduct is the restriction of an interpretation structure to a smaller signature. For every formula φ in $L(C)$ and each assignment α, the denotations of φ in the reduct and in $\mathcal{B}'$ are equal.

Lemma 3.1.26 *Let C, C' be signatures such that $C \leq C'$, $\mathcal{B}'$ an interpretation structure over C' and α' an assignment over $\mathcal{B}'$. Then*

$$[\![\varphi]\!]^{\alpha'}_{\mathcal{B}'|_C} = [\![\varphi]\!]^{\alpha'}_{\mathcal{B}'}$$

for every $\varphi \in L(C)$.

Proof. The result is easily established using induction. ◁

We now introduce the notion of fibring of interpretation systems.

Definition 3.1.27 The *fibring of two interpretation systems* $I' = \langle C', \mathcal{A}' \rangle$ and $I'' = \langle C'', \mathcal{A}'' \rangle$ is the interpretation system

$$I' \cup I'' = \langle C' \cup C'', \mathcal{A} \rangle$$

where $\mathcal{A}$ is the class of all interpretation structures $\mathcal{B}$ over $C' \cup C''$ such that $\mathcal{B}|_{C'} \in \mathcal{A}'$ and $\mathcal{B}|_{C''} \in \mathcal{A}''$. ▽

Remark 3.1.28 The last definition deserves some comments. Let

$$I' \cup I'' = \langle C' \cup C'', \mathcal{A} \rangle$$

be the fibring of the interpretation systems $I' = \langle C', \mathcal{A}' \rangle$ and $I'' = \langle C'', \mathcal{A}'' \rangle$.

Firstly, note that an interpretation structure in $\mathcal{A}$ should have the same truth-values and the same order relation as an interpretation structure in each of the components.

Moreover, each structure $\mathcal{B}$ in $\mathcal{A}$ can be seen as encompassing two structures: one belonging to $\mathcal{A}'$ (the structure $\mathcal{B}|_{C'}$) and the other belonging to $\mathcal{A}''$ (the structure $\mathcal{B}|_{C''}$). We say that $\mathcal{B}|_{C'}$ is the slice of $\mathcal{B}$ in $\mathcal{A}'$, whereas $\mathcal{B}|_{C''}$ is the slice of $\mathcal{B}$ in $\mathcal{A}''$. Clearly, not every structure in $\mathcal{A}'$ will necessarily appear as a slice $\mathcal{B}|_{C'}$ of a structure $\mathcal{B}$ in $\mathcal{A}$: just some structures in $\mathcal{A}'$ will persist in $\mathcal{A}$ under the form of a slice. The same arguments are valid for $\mathcal{A}''$. ▽

Observe that fibring extends fusion (see Section 1.2 of Chapter 1) at the semantic level for logics that are not only modal logics presented by Kripke structures.

Again we can define unconstrained and constrained fibring as for Hilbert calculi. Observe that if a constructor c of arity k is in both signatures C' and C'', then $\nu_k(c)(b_1, \ldots, b_k) = \nu'_k(c)(b_1, \ldots, b_k) = \nu''_k(c)(b_1, \ldots, b_k)$. That is, both structures should agree on the denotation of c.

We now show that fibring of interpretation systems preserves entailment, that is, semantic consequences holding in the components also hold in the fibring. We start by introducing a weakness relation between interpretation systems and proving that if an interpretation system I is weaker than other interpretation system I' then every entailment in I is also an entailment in I'.

Definition 3.1.29 The interpretation system $I = \langle C, \mathcal{A} \rangle$ is *weaker than* interpretation system $I' = \langle C', \mathcal{A}' \rangle$, written

$$I \leq I'$$

if $C \leq C'$ and $\mathcal{B}'|_C \in \mathcal{A}$ for every $\mathcal{B}' \in \mathcal{A}'$. ▽

Proposition 3.1.30 *Let $I = \langle C, \mathcal{A} \rangle$ and $I' = \langle C', \mathcal{A}' \rangle$ be interpretation systems such that $I \leq I'$. For every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$, if $\Gamma \vDash_I \varphi$ then $\Gamma \vDash_{I'} \varphi$.*

Proof. Consider $\mathcal{B}'$ in $\mathcal{A}'$ and an assignment α' over $\mathcal{B}'$ such that $[\![\gamma]\!]^{\alpha'}_{\mathcal{B}'} = \top$ for each $\gamma \in \Gamma$. Then $\mathcal{B}'|_C \in \mathcal{A}$ and, by Lemma 3.1.26, $[\![\gamma]\!]^{\alpha'}_{\mathcal{B}'|_C} = \top$ for each $\gamma \in \Gamma$. Since $\Gamma \vDash_I \varphi$, $[\![\varphi]\!]^{\alpha'}_{\mathcal{B}'|_C} = \top$ and, again by Lemma 3.1.26, $[\![\varphi]\!]^{\alpha'}_{\mathcal{B}'} = \top$. Hence, $\Gamma \vDash_{I'} \varphi$. ◁

The following result states that the original interpretation systems are weaker than the fibring. Therefore everything that is entailed by the components is also entailed by the fibring.

Proposition 3.1.31 *For every interpretation systems I' and I'', the following relationships hold: $I' \leq I' \cup I''$ and $I'' \leq I' \cup I''$.*

Proof. Immediate consequence of Definitions 3.1.27 and 3.1.29. ◁

Now we will see that the fibring of interpretation systems I' and I'' is minimal in the class of all interpretation systems that are stronger than I' and I''.

Proposition 3.1.32 *For every interpretation systems I, I' e I'', if $I' \leq I$ and $I'' \leq I$ then $I' \cup I'' \leq I$.*

Proof. Let $I' = \langle C', \mathcal{A}' \rangle$, $I'' = \langle C'', \mathcal{A}'' \rangle$, $I' \cup I'' = \langle C' \cup C'', \mathcal{A} \rangle$ and $I = \langle C_I, \mathcal{A}_I \rangle$. $C' \cup C'' \leq C_I$ since $C' \leq C_I$ and $C'' \leq C_I$. Given $\mathcal{B} \in \mathcal{A}_I$, $\mathcal{B}|_{C' \cup C''}$ is an interpretation structure over $C' \cup C''$. Moreover, $(\mathcal{B}|_{C' \cup C''})|_{C'} = \mathcal{B}|_{C'}$ and $(\mathcal{B}|_{C' \cup C''})|_{C''} = \mathcal{B}|_{C''}$. Since $\mathcal{B}|_{C'} \in \mathcal{A}'$ and $\mathcal{B}|_{C''} \in \mathcal{A}''$, $\mathcal{B}|_{C' \cup C''} \in \mathcal{A}$. ◁

From the two results above, the fibring $I' \cup I''$ is the supremum of I' and I'' with respect to the partial order of weakness.

We synthesize the properties of fibring of interpretation systems as follows:

- *homogeneous combination mechanism at the semantic level*: both original logics are presented by ordered algebras;
- *algorithmic combination of logics at the semantic level*: given the classes of ordered algebras for the original logics, we know how to define the class of ordered algebras for the fibring, but in many cases the given logics have to be pre-processed (that is, the ordered algebras have to be extracted from their given semantics);
- We will see that fibring of interpretation systems is also canonical.

Observe that fusion of logics is a particular case of fibring where the ordered algebras are powerset algebras induced by the Kripke structures. Hence the original normal modal logics in fusion induce two interpretation systems. The fibring of these interpretation systems corresponds to their fusion. Therefore, fusion of modal logics is also a universal construction when seen as a particular case of fibring.

Remark 3.1.33 Observe that fusion is also a universal construction even when we consider the original semantics. A Kripke interpretation system $\langle C, M \rangle$ is a pair such that C is a modal signature with $\neg$ and $\Rightarrow$ (shared connectives) and a finite number of $\Box$ connectives and M is a class of Kripke structures of the form $\langle W, \vec{R}, V \rangle$ (where $\vec{R}$ is a vector of relations one for each connective $\Box$ in C) .

A morphism $h : \langle C, M \rangle \to \langle C', M' \rangle$ between Kripke interpretation structures is a pair $\langle \overline{h}, \underline{h} \rangle$ where $\overline{h} : C \to C'$ is a signature morphism and $\underline{h} : M' \to M$ is a map such that $\underline{h}(\langle W', \vec{R}', V' \rangle) = \langle W', \vec{R}'|_C, V' \rangle$ where $\vec{R}'|_C$ is a sub-vector of $\vec{R}'$ restricted to the $\Box$ connectives in C.

Kripke interpretation structures and their morphisms constitute a category (observe that the morphisms we consider here are somehow more restricted than p-morphisms, see [24]). Fusion of modal logics is a pushout in this category. ▽

Example 3.1.34 Consider the modal signature C presented in Example 2.1.4 and the interpretation systems $I_{\mathbf{S4}} = \langle C, \mathcal{A}_{\mathbf{S4}} \rangle$ and $I_{\mathbf{B}} = \langle C, \mathcal{A}_{\mathbf{B}} \rangle$, corresponding respectively to modal logics **S4** and **B** (recall Example 3.1.20). Their fibring, sharing every constructor including the modality, is the interpretation system $I_{\mathbf{S4}} \cup I_{\mathbf{B}} = \langle C, \mathcal{A} \rangle$ where $\mathcal{A}$ is the class of all interpretations structures $\mathcal{B}$ over

C such that $\mathcal{B}|_{C'}$ is in $\mathcal{A}_{\mathbf{S4}}$ and $\mathcal{B}|_{C''}$ is in $\mathcal{A}_{\mathbf{B}}$. In other words, $\mathcal{A}$ is the class of all interpretation structures induced by Kripke frames whose accessibility relation is reflexive, transitive and symmetric. Thus, the accessibility relations of the Kripke frames of the fibring are equivalence relations, and so the resulting interpretation system corresponds to modal logic **S5**. The same result is obtained if we consider Kripke structures for **S4** and **B**. $\triangledown$

Example 3.1.35 Consider the interpretation systems $I_{\mathbf{S4}} = \langle C', \mathcal{A}_{\mathbf{S4}} \rangle$ and $I_{\mathbf{B}} = \langle C'', \mathcal{A}_{\mathbf{B}} \rangle$, where $C'_1 = \{\neg', \Box'\}$, $C'_2 = \{\Rightarrow'\}$, $C''_1 = \{\neg'', \Box''\}$, $C''_2 = \{\Rightarrow''\}$, corresponding respectively to modal logics **S4** and **B** (recall Example 3.1.20). Their fibring, sharing the propositional constructors but not the modality, is the interpretation system $I_{\mathbf{S4}} \cup I_{\mathbf{B}} = \langle C, \mathcal{A} \rangle$ such that:

- $C_1 = \{\neg, \Box', \Box''\}$ and $C_2 = \{\Rightarrow\}$;
- $\mathcal{A}$ is the class of all interpretations structures $\mathcal{B}$ over C such that $\mathcal{B}|_{C'}$ is in $\mathcal{A}_{\mathbf{S4}}$ and $\mathcal{B}|_{C''}$ is in $\mathcal{A}_{\mathbf{B}}$.

In other words, $\mathcal{A}$ is the class of all interpretation structures induced by bi-modal frames with a reflexive and transitive relation R' and a symmetric relation R''. Thus, we get the fusion of the two modal logics [113]. $\triangledown$

Example 3.1.36 Consider interpretation systems similar to the ones presented in Examples 3.1.17 and 3.1.19 for classical and intuitionistic logics, but where the signatures, respectively C' and C'', include different connectives for negation and implication, for instance $\neg'$ and $\Rightarrow'$ in C' and $\neg''$ and $\Rightarrow''$ in C''. Let I be the interpretation system resulting from their fibring. So, for each interpretation structure $\mathcal{B}$ in I, $\mathcal{B}|_{C'}$ is a Boolean algebra and $\mathcal{B}|_{C''}$ is a Heyting algebra. As the carrier set of both reducts coincide, $\mathcal{B}|_{C''}$ will also be a Boolean algebra and so I is an interpretation system for classical logic presented with a different signature. The intuitionistic part of I is then lost through the fibring or, using the terminology of Remark 3.1.28, the only Heyting algebras that persist in I are the Boolean algebras. This phenomenon of fibring collapsing was firstly described in [82] and [106]. Chapter 8 introduces the modulated fibring as a solution to this collapsing phenomenon. $\triangledown$

Remark 3.1.37 Fibring of interpretation systems can be characterized as an universal construction in the category of interpretation systems.

Recall again the signature morphisms presented in Remark 2.1.10. An interpretation system morphism

$$h : \langle C, \mathcal{A} \rangle \to \langle C', \mathcal{A}' \rangle$$

is a signature morphism $h : C \to C'$ such that, for every interpretation structure $\mathcal{B}' = \langle B', \leq', \nu', \top' \rangle$ in $\mathcal{A}'$, the reduct $\mathcal{B}'|_h = \langle B', \leq', \nu' \circ h, \top' \rangle$ is an interpretation structure in $\mathcal{A}$.

Interpretation systems and their morphisms, with composition and identities as in **Sig**, constitute the category **Int**. The category **Int** is (small) cocomplete.

The fibring $I' \cup I''$ of $I' = \langle C', \mathcal{A}' \rangle$ and $I'' = \langle C'', \mathcal{A}'' \rangle$ is a pushout of the morphisms

$$h' : \langle C' \cap C'', IntS(C' \cap C'') \rangle \to \langle C', \mathcal{A}' \rangle$$

and

$$h'' : \langle C' \cap C'', IntS(C' \cap C'') \rangle \to \langle C'', \mathcal{A}'' \rangle$$

where h' and h'' are the signature inclusion morphisms and $IntS(C' \cap C'')$ is the class of all the interpretations structures over $C' \cap C''$ (see Figure 3.1).

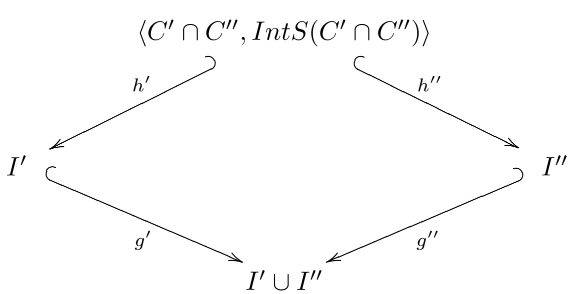


Figure 3.1: Fibring of interpretation systems as a pushout in **Int**

The particular case of unconstrained fibring just corresponds to the coproduct of I' and I''. ∇

Example 3.1.38 Recall the interpretation systems

$$I_{\mathbf{CPL}} = \langle C^{\mathbf{CPL}}, \mathcal{A}_{\mathbf{CPL}} \rangle$$

and

$$I_{\mathbf{S4}} = \langle C^{\mathbf{S4}}, \mathcal{A}_{\mathbf{S4}} \rangle$$

for classical logic and modal logic **S4** introduced in Examples 3.1.17 and 3.1.20.

The constrained fibring of $I_{\mathbf{CPL}}$ and $I_{\mathbf{S4}}$ (see Figure 3.2) is the interpretation system

$$I_{\mathbf{CPL}} \cup I_{\mathbf{S4}} = \langle C^{\mathbf{CPL}} \cup C^{\mathbf{S4}}, \mathcal{A} \rangle$$

where $\mathcal{A}$ is the class of interpretation structures $\mathcal{B}$ over $C^{\mathbf{CPL}} \cup C^{\mathbf{S4}}$ such that $\mathcal{B}|_{C^{\mathbf{CPL}}} \in \mathcal{A}_{\mathbf{CPL}}$ and $\mathcal{B}|_{C^{\mathbf{S4}}} \in \mathcal{A}_{\mathbf{S4}}$ for every $\mathcal{B} \in \mathcal{A}$.

Note that $C^{\mathbf{CPL}} \leq C^{\mathbf{S4}}$, thus $C^{\mathbf{CPL}} \cap C^{\mathbf{S4}} = C^{\mathbf{CPL}}$ and $C^{\mathbf{CPL}} \cup C^{\mathbf{S4}} = C^{\mathbf{S4}}$, thus $\mathcal{A}$ is the class $\mathcal{A}_{\mathbf{S4}}$ of all the interpretation structures induced by Kripke frames whose accessibility relation is reflexive and transitive.

It is worth noting that only some structures in $\mathcal{A}_{\mathbf{CPL}}$ persist in $\mathcal{A}$ as a slice of a structure (see Remark 3.1.28). The Boolean algebras that persist in $\mathcal{A}$ as slices are just the Boolean algebras of the form $\wp W$ for a non-empty set W. ∇

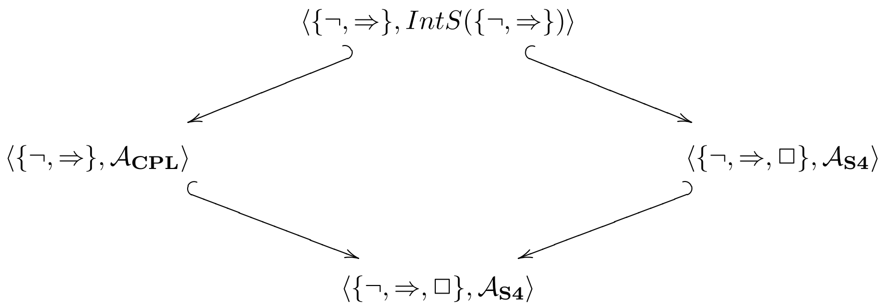


Figure 3.2: Fibring of $I_{\mathbf{CPL}}$ and $I_{\mathbf{S4}}$

Example 3.1.39 Consider now the interpretation system

$$I_{\mathit{2}} = \langle C^{\mathbf{CPL}}, \{\mathcal{B}_{\mathit{2}}\}\rangle$$

for classical logic introduced in Examples 3.1.17 and let $I_{\mathbf{S4}}$ be again the interpretation system $\langle C^{\mathbf{S4}}, \mathcal{A}_{\mathbf{S4}}\rangle$ for modal logic **S4**.

Reasoning as in Example 3.1.38 in the constrained fibring $I_{\mathit{2}} \cup I_{\mathbf{S4}}$, the class of structures is just a four-elements set

$$\mathcal{A} = \{\mathcal{B}^1_{\mathit{2}}, \mathcal{B}^2_{\mathit{2}}, \mathcal{B}^3_{\mathit{2}}, \mathcal{B}^4_{\mathit{2}}\}.$$

Each structure $\mathcal{B}^i_{\mathit{2}}$ is obtained from $\mathcal{B}_{\mathit{2}}$ just by adding a reflexive and transitive accessibility relation over 2 (observe that there are only four such relations over 2). Thus, the structures $\mathcal{B}^i_{\mathit{2}}$ (for $i = 1, \ldots, 4$) are the only structures of $\mathcal{A}_{\mathbf{S4}}$ that persist in the fibring. The corresponding diagram in the category **Int** is displayed in Figure 3.3. ∇

Example 3.1.40 Consider again the Examples 3.1.38 and 3.1.39, but now including the propositional symbols $\mathbb{P}$ (as constants) in the signatures. There are two possibilities: $\mathbb{P}$ can be shared or not. In order to keep the things simple, assume that $\mathbb{P}$ is shared (the results will not change essentially). Thus, the constrained fibring $\widehat{I}_{\mathbf{CPL}} \cup \widehat{I}_{\mathbf{S4}}$ while sharing the propositional symbols of $\widehat{I}_{\mathbf{CPL}}$ is displayed in Figure 3.4 where $\widehat{\mathcal{A}}_{\mathbf{CPL}}$ and $\widehat{\mathcal{A}}_{\mathbf{S4}}$ are as in Examples 3.1.18 and 3.1.20, respectively.

On the other hand, the constrained fibring $\widehat{I}_{\mathit{2}} \cup \widehat{I}_{\mathbf{S4}}$ while sharing the propositional symbols of $\widehat{I}_{\mathit{2}}$ can be described by the diagram in Figure 3.5. Observe that

$$\widehat{\mathcal{A}}_{\mathit{2}} = \{\mathcal{B}^V_{\mathit{2}} : V \in 2^{\mathbb{P}}\}$$

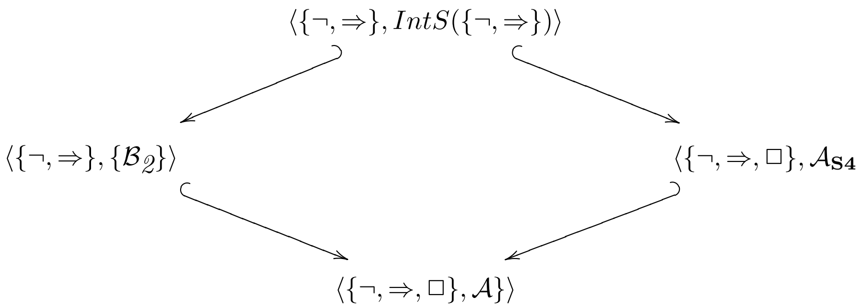


Figure 3.3: Fibring of I_{2} and $I_{\mathbf{S4}}$

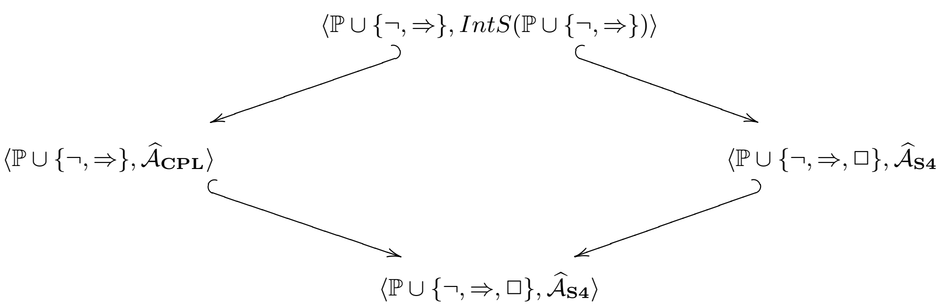


Figure 3.4: Fibring of $\widehat{I}_{\mathbf{CPL}}$ and $\widehat{I}_{\mathbf{S4}}$

where $\mathcal{B}_{2}^{V}$ is the interpretation structure induced by the Boolean algebra 2 and a valuation $V : \mathbb{P} \to 2$, and

$$\mathcal{A} = \bigcup_{V \in 2^{\mathbb{P}}} \{\mathcal{B}_{2}^{V,1}, \mathcal{B}_{2}^{V,2}, \mathcal{B}_{2}^{V,3}, \mathcal{B}_{2}^{V,4}\},$$

where $\mathcal{B}_{2}^{V,i}$ is obtained from $\mathcal{B}_{2}^{V}$ just by adding a reflexive and transitive accessibility relation over 2.

Thus, the unique interpretation structures of $\widehat{\mathcal{A}}_{\mathbf{S4}}$ that persist in the fibring are those of the form $\mathcal{B}_{2}^{V,i}$. $\triangledown$

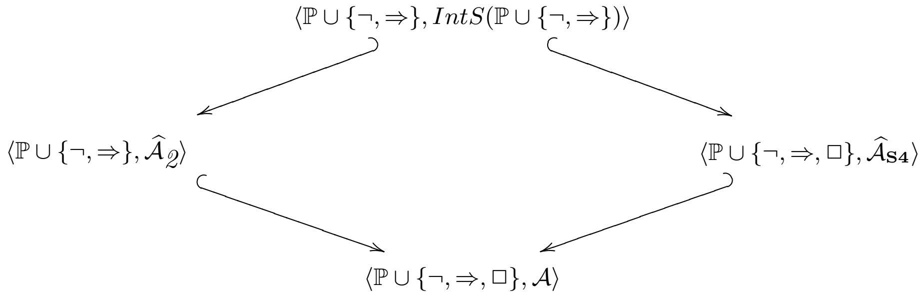


Figure 3.5: Fibring of $\widehat{I}_{\mathcal{2}}$ and $\widehat{I}_{\mathbf{S4}}$

3.2 Logic systems

Logic systems put together deductive and semantic aspects. They provide the right setting to present the notions of soundness and completeness. We start with the simple case where no distinction is made between local and global rules.

Definition 3.2.1 A *logic system* is a tuple

$$L = \langle C, \mathcal{A}, R\rangle$$

where $\langle C, \mathcal{A}\rangle$ is an interpretation system and $\langle C, R\rangle$ is a Hilbert calculus. $\triangledown$

Given the logic system $L = \langle C, \mathcal{A}, R\rangle$, we denote by $H(L)$ the Hilbert calculus $\langle C, R\rangle$ and we denote by $I(L)$ the interpretation system $\langle C, \mathcal{A}\rangle$. We write $\Gamma \vdash_L \varphi$ whenever $\Gamma \vdash_{H(L)} \varphi$ and write $\Gamma \vDash_L \varphi$ whenever $\Gamma \vDash_{I(L)} \varphi$.

Example 3.2.2 A logic system corresponding to classical logic is $\langle C, \mathcal{A}, R\rangle$ where $\langle C, R\rangle$ is the Hilbert calculus presented in Example 2.2.3 and $\langle C, \mathcal{A}\rangle$ is the interpretation system for classical logic presented in Example 3.1.17.

Similarly, from the Hilbert calculus presented in Example 2.2.6 and the interpretation system for intuitionistic logic presented in Example 3.1.19, we get a logic system corresponding to intuitionistic logic. $\triangledown$

Example 3.2.3 A logic system $\langle C, \mathcal{A}, R\rangle$, corresponding to normal modal logic **K**, is such that $\langle C, R\rangle$ is the Hilbert calculus presented in Example 2.2.4 and $\langle C, \mathcal{A}\rangle$ is the interpretation system for **K** induced by Kripke structures as presented in Example 3.1.20. From the corresponding Hilbert calculi and interpretation systems presented above, we get logic systems for modal logics **S4** and **B** in a similar way. $\triangledown$

The notion of soundness relates derivations to entailments: in a sound logic system, if we derive φ from Γ then Γ entails φ. Conversely, the notion of completeness relates entailments to derivations: in a complete logic system, if Γ entails φ then we can derive φ from Γ.

Definition 3.2.4 A logic system $L = \langle C, \mathcal{A}, R\rangle$ is said to be

- *sound* if $\Gamma \vDash_L \varphi$ whenever $\Gamma \vdash_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$;
- *complete* if $\Gamma \vdash_L \varphi$ whenever $\Gamma \vDash_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$. $\triangledown$

As expected, the fibring of two logic systems corresponds to the simultaneous fibring of their interpretation systems and their Hilbert calculi.

Definition 3.2.5 Let $L' = \langle C', \mathcal{A}', R'\rangle$ and $L'' = \langle C'', \mathcal{A}'', R''\rangle$ be logic systems. The *fibring* of L' and L'' is the logic system

$$L' \cup L'' = \langle C, \mathcal{A}, R\rangle$$

where $C = C' \cup C''$, $\mathcal{A}$ is the class of all interpretation structures $\mathcal{B}$ over $C' \cup C''$ such that $\mathcal{B}|_{C'} \in \mathcal{A}'$ and $\mathcal{B}|_{C''} \in \mathcal{A}''$, and $R = R' \cup R''$. $\triangledown$

Example 3.2.6 Recall the Hilbert calculi

$$H_{\mathbf{S4}} = \langle C, R_{\mathbf{S4}}\rangle \quad \text{and} \quad H_{\mathbf{B}} = \langle C, R_{\mathbf{B}}\rangle$$

for modal logics **S4** and **B**, respectively, presented in Example 2.2.14. Recall also the respective interpretation systems

$$I_{\mathbf{S4}} = \langle C, \mathcal{A}_{\mathbf{S4}}\rangle \quad \text{and} \quad I_{\mathbf{B}} = \langle C, \mathcal{A}_{\mathbf{B}}\rangle$$

introduced in Example 3.1.20. Let

$$L_{\mathbf{S4}} = \langle C, \mathcal{A}_{\mathbf{S4}}, R_{\mathbf{S4}}\rangle \quad \text{and} \quad L_{\mathbf{B}} = \langle C, \mathcal{A}_{\mathbf{B}}, R_{\mathbf{B}}\rangle$$

be the resulting logic systems.

The fibring $L_{\mathbf{S4}} \cup L_{\mathbf{B}}$ of $L_{\mathbf{S4}}$ and $L_{\mathbf{B}}$ is the logic system

$$\langle C, \mathcal{A}_{\mathbf{S5}}, R_{\mathbf{S5}}\rangle$$

where $\mathcal{A}_{\mathbf{S5}}$ is the class of structures for modal logic **S5** as described in Example 3.1.20, and $R_{\mathbf{S5}}$ the set of inference rules of the Hilbert calculus for **S5** as mentioned in Example 2.2.14. $\triangledown$

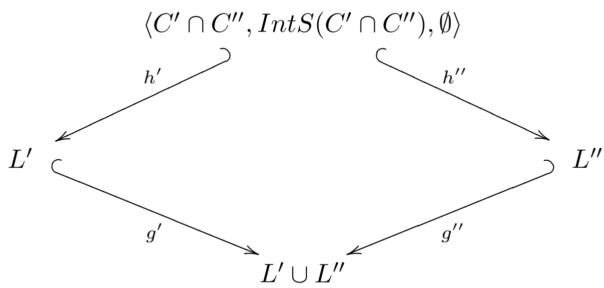


Figure 3.6: Fibring of logic systems as a pushout in **Log**

Remark 3.2.7 As expected, fibring of logic systems can be characterized as an universal construction in the category of logic systems.

A logic system morphism $h : \langle C, \mathcal{A}, R\rangle \to \langle C', \mathcal{A}', R'\rangle$ is a signature morphism $h : C \to C'$ such that $h : \langle C, \mathcal{A}\rangle \to \langle C', \mathcal{A}'\rangle$ is a morphism in the category **Int** and $h : \langle C, R\rangle \to \langle C', R'\rangle$ is a morphism in the category **Hil**. Logic systems and their morphisms, with composition and identities as in **Sig**, constitute the category **Log**. The category **Log** is (small) cocomplete.

The fibring of the logic systems $L' = \langle C', \mathcal{A}', R'\rangle$ and $L'' = \langle C'', \mathcal{A}'', R''\rangle$ is a pushout of the morphisms

$$h' : \langle C' \cap C'', IntS(C' \cap C''), \emptyset\rangle \to \langle C', \mathcal{A}', R'\rangle$$

and

$$h'' : \langle C' \cap C'', IntS(C' \cap C''), \emptyset\rangle \to \langle C'', \mathcal{A}'', R''\rangle$$

where h' and h'' are the signature inclusion morphisms and $IntS(C' \cap C'')$ is the class of all interpretations structures over $C' \cap C''$ (see Figure 3.6).

The particular case of unconstrained fibring just corresponds to the coproduct of L' and L''.

Soundness and completeness of logic systems correspond to the existence of suitable consequence system morphisms. The logic system $L = \langle C, \mathcal{A}, R\rangle$ is sound if the identity signature morphism $id_C : C \to C$ is a consequence system morphism $id_C : \langle C, \vdash_{H(L)}\rangle \to \langle C, \vDash_{I(L)}\rangle$. The logic system is complete if id_C is a consequence system morphism $id_C : \langle C, \vDash_{I(L)}\rangle \to \langle C, \vdash_{H(L)}\rangle$. ▽

Example 3.2.8 The fibring defined in Example 3.2.6 can be described in Figure 3.7 where $C = \{\neg, \Rightarrow, \Box\}$.

The logic system $L_{\mathbf{S5}} = \langle C, A_{\mathbf{S5}}, R_{\mathbf{S5}}\rangle$ is a logic system for modal logic **S5** and then

$$L_{\mathbf{S5}} = L_{\mathbf{S4}} \cup L_{\mathbf{B}}.$$

This shows that modal logic **S5** is obtained by splicing **S4** and **B** (see Figure 3.8 and recall Chapter 1). ▽

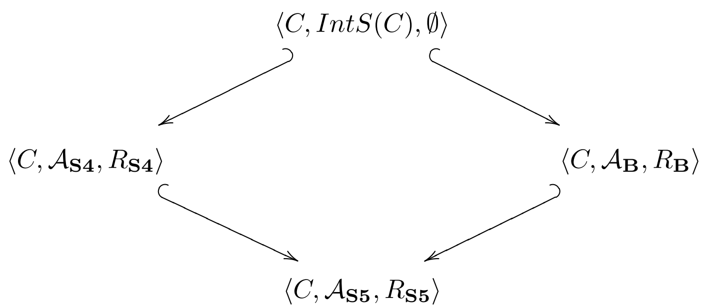


Figure 3.7: Fibring of $L_{\mathbf{S4}}$ and $L_{\mathbf{B}}$

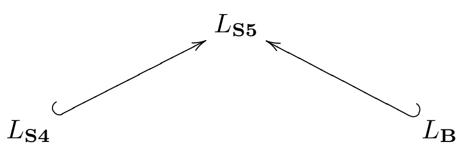


Figure 3.8: Splicing of **S4** and **B**

3.3 Preservation results

Herein, we continue the study initiated in Section 2.3 of Chapter 2 of transference results from the component logics to the logic resulting from their fibring. In this section we study the preservation of soundness and completeness properties.

3.3.1 Global and local entailment

Recall the distinction between global and local derivations discussed in Subsection 2.3.1 of Chapter 2. We now discuss the semantic counterpart of these notions: global and local entailments.

The distinction between these two kinds of entailment is easily motivated using Kripke semantics. We say that the set of formulas Γ globally entails a formula φ if, for every Kripke structure, whenever all the formulas in Γ are true in all worlds so is φ. This contrasts with local entailment, where Γ locally entails φ if, for each world of each Kripke structure, whenever all the formulas in Γ are true in the given world so is φ. As in the syntactic case, local entailment always implies global entailment, but the converse does not hold. For instance, φ globally entails $\Box\varphi$, but φ does not necessarily locally entails $\Box\varphi$.

The difference between global and local entailments is closed related to the semantic counterpart of the metatheorem of deduction. In fact, considering for instance the modal logic **K**, we cannot conclude that Γ entails $(\psi \Rightarrow \varphi)$ whenever

$\Gamma \cup \{\psi\}$ entails φ, if global entailment is considered. In fact, ψ globally entails $\Box\psi$, but, from a Kripke structure with only two worlds w_1 and w_2, where w_2 is accessible from w_1 and ψ only holds in w_1, it is easy to conclude that $(\psi \Rightarrow (\Box\psi))$ does not hold in w_1. Thus, the semantic version of the metatheorem of deduction does not hold as long as global entailment is considered. But there is no problem when considering local entailment, since if $\Gamma \cup \{\psi\}$ locally entails φ then Γ locally entails $(\psi \Rightarrow \varphi)$.

We now make more precise the notions of global entailment and local entailment in our setting.

Definition 3.3.1 Let $I = \langle C, \mathcal{A} \rangle$ be an interpretation system.

A formula $\varphi \in L(C)$ is *globally entailed* by $\Gamma \subseteq L(C)$ in I if it is entailed by Γ in the sense of Definition 3.1.22. We will write

$$\Gamma \vDash_I^g \varphi$$

to denote that φ is globally entailed by Γ in I.

A formula $\varphi \in L(C)$ is *locally entailed* by $\Gamma \subseteq L(C)$ if, for every interpretation structure $\mathcal{B} = \langle B, \leq, \nu, \top \rangle$ in I, assignment α over $\mathcal{B}$ and $b \in B$, if $b \leq [\![\gamma]\!]_{\mathcal{B}}^{\alpha}$ for each $\gamma \in \Gamma$ then $b \leq [\![\varphi]\!]_{\mathcal{B}}^{\alpha}$. We will write

$$\Gamma \vDash_I^\ell \varphi$$

to denote that φ is locally entailed by Γ in I. ∇

Observe that global entailment just corresponds to the notion of entailment we had before. When in the interpretation structures of an interpretation system I it is meaningful to refer to the meet operation $\sqcap$, as, for instance, in the structures induced by Boolean or Heyting algebras, we can give an equivalent definition of local entailment when Γ is a finite set: $\Gamma \vDash_I^\ell \varphi$ if, for every $\mathcal{B}$ in I and assignment α over $\mathcal{B}$, $(\sqcap_{\gamma\in\Gamma}[\![\gamma]\!]_{\mathcal{B}}^{\alpha}) \leq [\![\varphi]\!]_{\mathcal{B}}^{\alpha}$.

Local entailment implies global entailment, but global entailment does not necessarily implies local entailment. We denote by $\Gamma^{\vDash_I^g}$ and $\Gamma^{\vDash_I^\ell}$ respectively the set of formulas that are globally entailed from Γ in I and the set of formulas that are locally entailed from Γ in I. As noted above, $\Gamma^{\vDash_I^\ell} \subseteq \Gamma^{\vDash_I^g}$ for every Γ.

Example 3.3.2 Consider the interpretation system I for modal logic **K** that includes all the interpretation structures $\mathcal{B}$ induced by Kripke frames $\langle W, R \rangle$, and let α be an assignment over $\mathcal{B}$. Recall that $\top = W$. If $[\![\xi_1]\!]_{\mathcal{B}}^{\alpha} = \top$ then

$$[\![\Box\xi_1]\!]_{\mathcal{B}}^{\alpha} = \{w : \text{ if } wRv \text{ then } v \in W\} = \top.$$

Hence $\xi_1 \vDash_I^g \Box\xi_1$.

Consider now the interpretation structure $\mathcal{B}$ induced by the Kripke frame presented in Example 3.1.15 and the assignment α over $\mathcal{B}$ such that $\alpha(\xi_1) = \{w_1\}$. Then $[\![\xi_1]\!]_{\mathcal{B}}^{\alpha} = \{w_1\}$ and $[\![\Box\xi_1]\!]_{\mathcal{B}}^{\alpha} = \emptyset$. Hence

$$\{w_1\} \leq [\![\xi_1]\!]_{\mathcal{B}}^{\alpha} \text{ but } \{w_1\} \not\leq [\![\Box\xi_1]\!]_{\mathcal{B}}^{\alpha}.$$

We conclude that ξ_1 does not locally entail $\Box\xi_1$ in I. ▽

Remark 3.3.3 Note that in every interpretation system $I = \langle C, \mathcal{A}\rangle$ local entailment also induces a structural consequence system, the consequence system $\langle C, \vDash^{\ell}_{I}\rangle$. The proof is similar to the one presented for Proposition 3.1.24.

Clearly, whenever $I \leq I'$, if $\Gamma \vDash^g_I \varphi$ then $\Gamma \vDash^g_{I'} \varphi$, since the notion of global entailment is just the notion of entailment we had before. The result also holds with local entailment: if $\Gamma \vDash^\ell_I \varphi$ then $\Gamma \vDash^\ell_{I'} \varphi$. The proof is similar to the one presented for Proposition 3.1.30. ▽

Clearly, we can also extend logic systems to include both a set of global inference rules and a set of local inference rules, that is,

$$L = \langle C, \mathcal{A}, R_g, R_\ell\rangle$$

where $\langle C, R_g, R_\ell\rangle$ is a Hilbert calculus with careful reasoning (recall Definition 2.3.1). Global soundness and global completeness as well as local soundness and local completeness are defined as expected.

Definition 3.3.4 A logic system $L = \langle C, \mathcal{A}, R_g, R_\ell\rangle$ is said to be

- *globally sound* if $\Gamma \vDash^g_L \varphi$ whenever $\Gamma \vdash^g_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$;
- *locally sound* if $\Gamma \vDash^\ell_L \varphi$ whenever $\Gamma \vdash^\ell_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$;
- *globally complete* if $\Gamma \vdash^g_L \varphi$ whenever $\Gamma \vDash^g_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$;
- *locally complete* if $\Gamma \vdash^\ell_L \varphi$ whenever $\Gamma \vDash^\ell_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$.

A logic system is said to be *sound* (*complete*) if it is simultaneously globally and locally sound (complete). ▽

As expected, the properties of Hilbert calculi referred to in Section 2.3.2 of Chapter 2 can be also seen as properties of logic systems. For instance, a logic system L is said to have implication if the Hilbert calculus $H(L)$ has implication, L has *MTC* if $H(L)$ has *MTC*, and so on.

In the same vein, the fibring of these extended logic systems can be defined in the natural way.

Definition 3.3.5 Let $L' = \langle C', \mathcal{A}', {R_g}', {R_\ell}'\rangle$ and $L'' = \langle C'', \mathcal{A}'', {R_g}'', {R_\ell}''\rangle$ be two logic systems. The *fibring* of L' and L'' is the logic system

$$L' \cup L'' = \langle C, \mathcal{A}, R_g, R_\ell\rangle$$

where $C = C' \cup C''$, $\mathcal{A}$ is the class of all interpretation structures $\mathcal{B}$ over $C' \cup C''$ such that $\mathcal{B}|_{C'} \in \mathcal{A}'$ and $\mathcal{B}|_{C''} \in \mathcal{A}''$, $R_g = {R_g}' \cup {R_g}''$ and $R_\ell = {R_\ell}' \cup {R_\ell}''$. ▽

Again fibring of logic systems is a pushout in the relevant category. In the sequel, when we refer to a logic system we assume that they include a set of global inference rules and a set of local inference rules.

3.3.2 Soundness

In this section we study the preservation by fibring of soundness of logic systems. We prove that the fibring of globally sound logic systems is also a globally sound logic system. Similarly with respect to local soundness.

To begin with, we have to introduce the notion of globally sound inference rule and locally sound inference rule.

Definition 3.3.6 Let $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ be a logic system. An inference rule $\langle \Delta, \varphi \rangle \in R_g \cup R_\ell$ is *globally sound* in L if $\Delta \vDash^g_L \varphi$ and it is *locally sound* in L if $\Delta \vDash^\ell_L \varphi$. ∇

As expected, a logic system is globally sound whenever all its inference rules are globally sound. It is locally sound whenever global inference rules are globally sound and local inference rules are locally sound.

Theorem 3.3.7 *Let $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ be a logic system.*

1. *If every inference rule in R_g is globally sound then L is globally sound.*

2. *If every inference rule in R_g is globally sound and every inference rule in R_ℓ is locally sound then L is locally sound.*

Proof. The proof follows the usual steps.

1. We prove by induction on the length of a global derivation of φ from Γ that

$$\Gamma \vDash^g_L \varphi$$

whenever $\Gamma \vdash^g_L \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$.

Base: If φ has a derivation of length one from Γ, then $\varphi \in \Gamma$ or $\varphi = \sigma(\psi)$ for some axiom $\langle \emptyset, \psi \rangle$ in R_g and substitution σ. The first case is trivial. In the second case, consider $\mathcal{B}$ in $\mathcal{A}$ and an assignment α over $\mathcal{B}$. By Lemma 3.1.23,

$$[\![\sigma(\psi)]\!]^\alpha_\mathcal{B} = [\![\psi]\!]^{\alpha'}_\mathcal{B}$$

where α' is such that $\alpha'(\xi) = [\![\sigma(\xi)]\!]^\alpha_\mathcal{B}$. Since $\langle \emptyset, \psi \rangle$ is globally sound, $[\![\psi]\!]^\alpha_\mathcal{B} = \top$ for every interpretation structure $\mathcal{B}$ in $\mathcal{A}$ and assignment α over $\mathcal{B}$. Hence, $[\![\sigma(\psi)]\!]^\alpha_\mathcal{B} = \top$ and therefore $\Gamma \vDash^g_L \varphi$.

Step: If φ has a derivation of length n+1 from Γ, the only relevant case is $\varphi = \sigma(\psi)$ for some rule $\langle \{\delta_1, \ldots, \delta_k\}, \psi \rangle$ in R_g and substitution σ such that each $\sigma(\delta_i)$ occurs previously in the derivation. Consider $\mathcal{B}$ in $\mathcal{A}$ and α an assignment over $\mathcal{B}$ such that $[\![\gamma]\!]_{\mathcal{B}}^{\alpha} = \top$ for every $\gamma \in \Gamma$. By the induction hypothesis, $\Gamma \vDash_L^g \sigma(\delta_i)$ and therefore $[\![\sigma(\delta_i)]\!]_{\mathcal{B}}^{\alpha} = \top$, $i = 1, \ldots, k$. By Lemma 3.1.23,

$$[\![\delta_i]\!]_{\mathcal{B}}^{\alpha'} = [\![\sigma(\delta_i)]\!]_{\mathcal{B}}^{\alpha} = \top$$

$i = 1, \ldots, k$, with α' as above. Since the rule is globally sound, $[\![\psi]\!]_{\mathcal{B}}^{\alpha'} = \top$. Finally, by Lemma 3.1.23,

$$[\![\sigma(\psi)]\!]_{\mathcal{B}}^{\alpha} = \top.$$

Therefore $\Gamma \vDash_L^g \varphi$.

2. We prove by induction on the length of derivations that $\Gamma \vDash_L^\ell \varphi$ whenever $\Gamma \vdash_L^\ell \varphi$, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$.

Base: If φ has a derivation of length one from Γ, then $\varphi \in \Gamma$, $\varphi = \sigma(\psi)$ for some axiom $\langle \emptyset, \psi \rangle$ in R_ℓ and substitution σ or $\vdash_L^g \varphi$. The first case is trivial. In the other two cases, consider $\mathcal{B}$ in $\mathcal{A}$, with carrier set B, an assignment α over $\mathcal{B}$, and $b \in B$ such that $b \leq [\![\gamma]\!]_{\mathcal{B}}^{\alpha}$ for every $\gamma \in \Gamma$. In the second case, by Lemma 3.1.23,

$$[\![\sigma(\psi)]\!]_{\mathcal{B}}^{\alpha} = [\![\psi]\!]_{\mathcal{B}}^{\alpha'}$$

with α' such that $\alpha'(\xi) = [\![\sigma(\xi)]\!]_{\mathcal{B}}^{\alpha}$. Since $\langle \emptyset, \psi \rangle$ is locally sound, $b \leq [\![\psi]\!]_{\mathcal{B}}^{\alpha}$ for every interpretation structure $\mathcal{B}$ in $\mathcal{A}$, assignment α over $\mathcal{B}$ and b in the carrier set of $\mathcal{B}$. Hence, $b \leq [\![\sigma(\psi)]\!]_{\mathcal{B}}^{\alpha}$ and therefore $\Gamma \vDash_L^\ell \varphi$. Finally, if $\vdash_L^g \varphi$, since all inference rules in R_g are globally sound, from 1 we have $\vDash_L^g \varphi$, that is,

$$[\![\varphi]\!]_{\mathcal{B}}^{\alpha} = \top$$

for every interpretation structure $\mathcal{B}$ in $\mathcal{A}$ and assignment α over $\mathcal{B}$. Thus, $b \leq [\![\varphi]\!]_{\mathcal{B}}^{\alpha}$ and therefore $\Gamma \vDash_L^\ell \varphi$.

Step: If φ has a derivation of length n+1 from Γ, the only relevant case is $\varphi = \sigma(\psi)$ for some rule

$$\langle \{\delta_1, \ldots, \delta_k\}, \psi \rangle$$

in R_ℓ and substitution σ such that each $\sigma(\delta_i)$ occurs previously in the derivation. Consider $\mathcal{B}$ in $\mathcal{A}$, with carrier set B, an assignment α over $\mathcal{B}$ and $b \in B$ such that $b \leq [\![\gamma]\!]_{\mathcal{B}}^{\alpha}$ for every $\gamma \in \Gamma$. By the induction hypothesis, $\Gamma \vDash_L^\ell \sigma(\delta_i)$ and therefore

$$b \leq [\![\sigma(\delta_i)]\!]_{\mathcal{B}}^{\alpha}$$

$i = 1, \ldots, k$. Using Lemma 3.1.23,

$$b \leq [\![\delta_i]\!]_{\mathcal{B}}^{\alpha'} = [\![\sigma(\delta_i)]\!]_{\mathcal{B}}^{\alpha}$$

$i = 1, \ldots, k$, with α' as above. Since the rule is locally sound, $b \leq [\![\psi]\!]_{\mathcal{B}}^{\alpha'}$. Finally, by Lemma 3.1.23, $b \leq [\![\sigma(\psi)]\!]_{\mathcal{B}}^{\alpha}$. Therefore $\Gamma \vDash_L^\ell \varphi$. $\triangleleft$

We now prove that fibring preserves both global and local soundness.

Theorem 3.3.8 *The fibring of globally sound logic systems is also a globally sound logic system and the fibring of locally sound logic systems is also a locally sound logic system.*

Proof. Let $L' = \langle C', \mathcal{A}', {R_g}', {R_\ell}' \rangle$ and $L'' = \langle C'', \mathcal{A}'', {R_g}'', {R_\ell}'' \rangle$ be logic systems and $L' \cup L''$ their fibring. Note that

$$I(L') \leq I(L' \cup L'') \text{ and } I(L'') \leq I(L' \cup L'').$$

Assume that L' and L'' are globally sound. Let $\langle \Delta, \psi \rangle \in {R_g}'$. Clearly, $\Delta \vdash^g_{L'} \varphi$. Since L' is globally sound, $\Delta \vDash^g_{L'} \varphi$. Using Remark 3.3.3,

$$\Delta \vDash^g_{L' \cup L''} \varphi.$$

Hence $\langle \Delta, \varphi \rangle$ is globally sound in $L' \cup L''$. Similarly, each $\langle \Delta, \varphi \rangle \in {R_g}''$ is globally sound in $L' \cup L''$. Thus, every global rule of $L' \cup L''$ is globally sound in $L' \cup L''$ and therefore, by Proposition 3.3.7, $L' \cup L''$ is globally sound.

Assume now that L' and L'' are also locally sound. Let $\langle \Delta, \varphi \rangle \in {R_\ell}'$. Clearly, $\Delta \vdash^\ell_{L'} \varphi$. Since L' is locally sound, $\Delta \vDash^\ell_{L'} \varphi$. Using again Remark 3.3.3,

$$\Delta \vDash^\ell_{L' \cup L''} \varphi.$$

Hence $\langle \Delta, \varphi \rangle$ is locally sound in $L' \cup L''$. Similarly, each $\langle \Delta, \varphi \rangle \in {R_\ell}''$ is locally sound in $L' \cup L''$. Thus, every global rule of $L' \cup L''$ is globally sound in $L' \cup L''$ and every local rule of $L' \cup L''$ is locally sound in $L' \cup L''$. Therefore, by Proposition 3.3.7, $L' \cup L''$ is locally sound. ◁

Example 3.3.9 Recall the modal signatures C' and C'' presented in Example 2.1.8. Consider the logic system $L'_{\mathbf{S4}}$, corresponding to modal logic **S4**, whose signature is C', $H(L'_{\mathbf{S4}})$ is the Hilbert calculus with careful reasoning for **S4** similar to the one referred to in Example 2.3.3 and the class of interpretations structures in $I(L'_{\mathbf{S4}})$ includes all the interpretations structures induced by Kripke frames with reflexive and transitive accessibility relation.

Consider also the logic system $L''_{\mathbf{B}}$, corresponding to modal logic **B**, whose signature is C'', $H(L''_{\mathbf{B}})$ is the Hilbert calculus with careful reasoning for **B** similar to the one referred to in Example 2.3.3 and the class of interpretation structures in $I(L''_{\mathbf{B}})$ includes all the interpretation structures induced by Kripke frames with symmetric relation.

The logic systems $L'_{\mathbf{S4}}$ and $L''_{\mathbf{B}}$ are both globally and locally sound. Hence, the fibring $L'_{\mathbf{S4}} \cup L''_{\mathbf{B}}$ is also globally and locally sound and corresponds to the logic system of modal logic **S5**. ▽

Note that in larger logical universes things can be more complicated. As it is shown in Chapter 7, when fibring logic systems with quantifiers and using rules with side provisos (such as "provided that term t is free for variable x in formula ξ"), soundness is not always preserved.

3.3.3 Completeness

We now turn our attention to the preservation by fibring of global completeness of logic systems. To this end we establish sufficient properties for a logic system to be global complete and then show that fibring preserves these properties.

We first introduce the notions of full logic system and logic system with *verum*. Previous to this, we need to introduce a notion derived from Definition 3.3.1.

Definition 3.3.10 Let $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ be a logic system, $\langle \Delta, \varphi \rangle \in R_g \cup R_\ell$, and let $\mathcal{B} = \langle B, \leq, \nu, \top \rangle$ be an interpretation structure over C. Then:

- $\mathcal{B}$ *globally satisfies* $\langle \Delta, \varphi \rangle$ if $\Delta \vDash^g_{\langle C, \{\mathcal{B}\} \rangle} \varphi$.
- $\mathcal{B}$ *locally satisfies* $\langle \Delta, \varphi \rangle$ if $\Delta \vDash^\ell_{\langle C, \{\mathcal{B}\} \rangle} \varphi$. ▽

Definition 3.3.11 A logic system $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ is said to be *full* when $\mathcal{A}$ contains all interpretation structures over C that globally satisfy the inference rules in R_g and that locally satisfy the inference rules in R_ℓ. ▽

Example 3.3.12 Consider the logic system $L_{\mathbf{K}}$ such that $H(L_{\mathbf{K}})$ is the Hilbert calculus for normal modal logic **K** presented in Example 2.3.3, and $I(L)$ is the interpretation system for **K** obtained from Kripke frames presented in Example 3.1.20. $I(L_{\mathbf{K}})$ includes all the interpretation structures induced by Kripke frames and it is easy to prove that each of these structures locally satisfies the inference rules in R_ℓ and globally satisfies the ones in R_g. Observe that other structures can satisfy the inferences rules in $H(L_{\mathbf{K}})$.

Consider the logic system $L_{\mathbf{S4}}$ presented in Example 3.3.9. $I(L_{\mathbf{S4}})$ includes all the interpretation structures induced by Kripke frames with a reflexive and transitive accessibility relation. Those are in fact all the interpretation structures induced by Kripke frames that locally satisfies the inference rules in R_ℓ and globally satisfies the inference rules in R_g. Observe that other structures can satisfy the inferences rules in $H(L_{\mathbf{S4}})$. Similar remarks hold for the logic system $L_{\mathbf{B}}$ presented in Example 3.3.9. ▽

Definition 3.3.13 A logic system $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ is said to have *verum* when there is a formula $\varphi \in L(C)$ such that $\vdash^g_{H(L)} \varphi$ and $[\![\varphi]\!]^\alpha_{\mathcal{B}} = \top$ for every $\mathcal{B}$ in $\mathcal{A}$ and assignment α over $\mathcal{B}$. ▽

The formulas whose denotation is always $\top$, *verum* formulas, constitute the syntactical counterpart of the top element of the interpretation structures. Note that instead of requiring that $\vdash^g_{H(L)} \varphi$ we could have required that $\vdash^\ell_{H(L)} \varphi$ since the two conditions are equivalent.

Example 3.3.14 Let $L_{\mathbf{K}}$ be the logic system corresponding to normal modal logic **K** presented in Example 3.3.12. This logic system has *verum*. In fact, taking any formula φ in the language, we have that $(\varphi \Rightarrow \varphi)$ satisfies the conditions in Definition 3.3.13. The same applies to the other logic systems considered above. $\triangledown$

The technique for proving preservation of completeness is different from the ones that we used for proving preservation of interpolation (see Chapter 2) and preservation of soundness. In this case we start by proving sufficient conditions for completeness of a stand-alone logic system. Then we prove the preservation of those conditions, hence proving the completeness of the fibring.

We are now ready to state the completeness theorem for global reasoning. The proof uses a common Lindenbaum-Tarski construction (sometimes called simply Lindenbaum algebra; this construction is named for logicians Adolf Lindenbaum and Alfred Tarski). Recall the notion of Hilbert calculus with *MTC* given in Definition 2.3.29.

Theorem 3.3.15 *Every full logic system with MTC and* verum *is globally complete.*

Proof. Let $L = \langle C, \mathcal{A}, R_g, R_\ell \rangle$ be a full logic system and let $\gamma_v \in L(C)$ be a formula satisfying the *verum* conditions.

(i) Let Γ be a globally closed subset of $L(C)$. Our first goal is the construction of an interpretation structure for C based on Γ. We start by defining the following binary relation $\cong_\Gamma$ on $L(C)$:

$$\varphi \cong_\Gamma \psi \;\text{ if }\; \Gamma, \varphi \vdash_L^\ell \psi \text{ and } \Gamma, \psi \vdash_L^\ell \varphi.$$

It is easy to prove that $\cong_\Gamma$ is an equivalence relation and, since L has *MTC*, $\cong_\Gamma$ is also a congruence relation. We can then consider the interpretation structure

$$\mathcal{B}_\Gamma = (B_\Gamma, \leq_\Gamma, \nu_\Gamma, \top_\Gamma)$$

where

- B_Γ is the quotient set $L(C)/\cong_\Gamma$;
- $[\varphi] \leq_\Gamma [\psi]$ if $\Gamma, \varphi \vdash_L^\ell \psi$;
- $\nu_\Gamma(c) = [c]$ for $c \in C_0$ and $\nu_\Gamma(c)([\varphi_1], \ldots, [\varphi_k]) = [c(\varphi_1, \ldots, \varphi_k)]$ for $c \in C_k$;
- $\top_\Gamma = [\gamma_v]$.

(ii) We now prove some useful properties of the interpretation structure $\mathcal{B}_\Gamma$ defined in (i).

(ii.a) Let α be the assignment over $\mathcal{B}_\Gamma$ such that $\alpha(\xi) = [\psi_\xi]$ for each $\xi \in \Xi$. It is easily proved by induction that, for every $\varphi \in L(C)$,

$$[\![\varphi]\!]^\alpha_{\mathcal{B}_\Gamma} = [\sigma(\varphi)]$$

where σ is the substitution such that $\sigma(\xi) = \psi_\xi$ for every $\xi \in \Xi$ (note that if $[\psi_\xi] = [\psi'_\xi]$ then $[\sigma(\varphi)] = [\sigma'(\varphi)]$ where $\sigma'(\xi) = \psi'_\xi$).

(ii.b) Consider an assignment α and a substitution σ as in (a). Let ι be the assignment on $\mathcal{B}_\Gamma$ such that $\iota(\xi) = [\xi]$ for every $\xi \in \Xi$. We prove that, for every $\varphi \in L(C)$,

$$[\![\sigma(\varphi)]\!]^\iota_{\mathcal{B}_\Gamma} = [\![\varphi]\!]^\alpha_{\mathcal{B}_\Gamma}.$$

In fact, using Lemma 3.1.23, $[\![\sigma(\varphi)]\!]^\iota_{\mathcal{B}_\Gamma} = [\![\varphi]\!]^{\alpha'}_{\mathcal{B}_\Gamma}$ where α' is the assignment such that $\alpha'(\xi) = [\![\sigma(\xi)]\!]^\iota_{\mathcal{B}_\Gamma}$ for every $\xi \in \Xi$. But $[\![\sigma(\xi)]\!]^\iota_{\mathcal{B}_\Gamma} = [\![\psi_\xi]\!]^\iota_{\mathcal{B}_\Gamma}$. Using (a), with $\alpha = \iota$, we get

$$[\![\psi_\xi]\!]^\iota_{\mathcal{B}_\Gamma} = [\sigma_\iota(\psi_\xi)]$$

where $\sigma_\iota(\xi) = \xi$ for every $\xi \in \Xi$. Since $\sigma_\iota(\psi_\xi) = \psi_\xi$, we have $[\![\psi_\xi]\!]^\iota_{\mathcal{B}_\Gamma} = [\psi_\xi]$. Hence, $\alpha'(\xi) = [\psi_\xi]$, for every $\xi \in \Xi$. Therefore, α' is α. We then conclude that $[\![\sigma(\varphi)]\!]^\iota_{\mathcal{B}_\Gamma} = [\![\varphi]\!]^\alpha_{\mathcal{B}_\Gamma}$.

(ii.c) Let ι be the assignment in (b). We now prove that, for every $\varphi \in L(C)$,

$$[\![\varphi]\!]^\iota_{\mathcal{B}_\Gamma} = \top_\Gamma \text{ if and only if } \varphi \in \Gamma.$$

Assuming that $\varphi \in \Gamma$, clearly $\Gamma, \gamma_v \vdash^\ell_L \varphi$. Since $\vdash^g_L \gamma_v$, also $\Gamma, \varphi \vdash^\ell_L \gamma_v$. Hence, $\varphi \cong_\Gamma \gamma_v$. Thus,

$$[\varphi] = [\gamma_v]$$

that is, using (a), $[\![\varphi]\!]^\iota_{\mathcal{B}_\Gamma} = \top_\Gamma$. Conversely, assuming that $[\![\varphi]\!]^\iota_{\mathcal{B}_\Gamma} = \top_\Gamma$, then $[\varphi] = [\gamma_v]$. Thus, we have

$$\Gamma, \gamma_v \vdash^\ell_L \varphi.$$

Since Γ is globally closed, $\gamma_v \in \Gamma$. Hence, $\Gamma \vdash^\ell_L \varphi$. Then also $\Gamma \vdash^g_L \varphi$ and therefore $\varphi \in \Gamma$.

(ii.d) Finally, we prove that $\mathcal{B}_\Gamma$ globally satisfies the inference rules in R_g and locally satisfies the inference rules in R_ℓ. Let $\langle \Delta, \varphi \rangle \in R_g$ and α an assignment over $\mathcal{B}_\Gamma$ such that $[\![\delta]\!]^\alpha_{\mathcal{B}_\Gamma} = \top_\Gamma$ for every $\delta \in \Delta$. But, using (b),

$$[\![\delta]\!]^\alpha_{\mathcal{B}_\Gamma} = [\![\sigma(\delta)]\!]^\iota_{\mathcal{B}_\Gamma}$$

for every $\delta \in \Delta$. Using (c), $\sigma(\delta) \in \Gamma$ for every $\delta \in \Delta$. Hence $\Gamma \vdash^g_L \sigma(\varphi)$ and, since Γ is globally closed, $\sigma(\varphi) \in \Gamma$. Using again (c) e (b),

$$[\![\sigma(\varphi)]\!]^\iota_{\mathcal{B}_\Gamma} = \top_\Gamma = [\![\varphi]\!]^\alpha_{\mathcal{B}_\Gamma}.$$

Hence $\mathcal{B}_\Gamma$ globally satisfies $\langle \Delta, \varphi \rangle$. Consider now $\langle \Delta, \varphi \rangle \in R_\ell$, α an assignment over $\mathcal{B}_\Gamma$ and $[\psi] \in B_\Gamma$ such that $[\psi] \leq_\Gamma [\![\delta]\!]^\alpha_{\mathcal{B}_\Gamma}$ for every $\delta \in \Delta$. Using (a),

$$[\psi] \leq_\Gamma [\sigma(\delta)]$$

for every $\delta \in \Delta$. As a consequence, $\Gamma, \psi \vdash^\ell_L \sigma(\delta)$ for every $\delta \in \Delta$. Therefore, $\Gamma, \psi \vdash^\ell_L \sigma(\varphi)$, that is, $[\psi] \leq_\Gamma [\sigma(\varphi)]$. Using (a) again, $[\psi] \leq_\Gamma [\![\varphi]\!]^\alpha_{\mathcal{B}_\Gamma}$. Hence $\mathcal{B}_\Gamma$ locally satisfies $\langle \Delta, \varphi \rangle$.

(iii) Finally, we prove the global completeness of L. Let $\Theta \subseteq L(C)$ and consider $\varphi \in L(C)$. We prove that if

$$\Theta \not\vdash^g_L \varphi \text{ then } \Theta \not\vDash^g_L \varphi.$$

Consider the interpretation structure $\mathcal{B}_\Gamma$, where $\Gamma = \Theta^{\vdash^g_L}$. Since L is full, by (ii.d) above, $\mathcal{B}_\Gamma$ is an interpretation structure in $\mathcal{A}$. Using (ii.c) above, for each $\gamma \in \Theta \subseteq \Gamma$, $[\![\gamma]\!]^\iota_{\mathcal{B}_\Gamma} = \top_\Gamma$. Since $\Theta \not\vdash^g_L \varphi$, $\varphi \notin \Gamma$. Using again (ii.c), $[\![\varphi]\!]^\iota_{\mathcal{B}_\Gamma} \neq \top_\Gamma$. Hence $\Theta \not\vDash^g_L \varphi$. ◁

Observe that the requirements of congruence and *verum* are quite weak and usually fulfilled by commonly used logic systems (including those mentioned above as examples). Furthermore, any complete logic system can be made full without changing its entailment. However, it should be observed that when adding all the possible models of a given logic it can be the case that some unexpected or exotic models can appear. On the other hand, if *verum* is not present, it can be conservatively added in congruent logic systems. But if the system at hand is not congruent, there is nothing we can do within the scope of the basic theory of fibring outlined here.

Note also that through a mild strengthening of the requirements of the theorem we can ensure finitary local completeness (see for instance [243]). A similar local and global completeness theorem is obtained in [282] without extra requirements for local reasoning but assuming a more complex semantics and using a Henkin construction.

Preservation of global completeness follows by adapting the technique originally proposed in [282], and capitalizing on the completeness result stated above. That is, when fibring two given logic systems that are full, congruent and with *verum* (and, therefore, globally complete) we shall try to obtain the global completeness of the result by identifying the conditions under which fullness, congruence and *verum* are preserved by fibring.

Theorem 3.3.16 *Fullness is preserved by fibring.*

Proof. Let

$$L' = \langle C', \mathcal{A}', R_g{}', R_\ell{}' \rangle \text{ and } L'' = \langle C'', \mathcal{A}'', R_g{}'', R_\ell{}'' \rangle$$

be full logic systems. Let $\mathcal{B}$ be an interpretation structure over $C' \cup C''$ that globally satisfies all the inference rules in $R_g{}' \cup R_g{}''$ and locally satisfies all the inference rules in $R_\ell{}' \cup R_\ell{}''$. We have to prove that $\mathcal{B}$ is in $\mathcal{A}$, that is,

$$\mathcal{B}|_{C'} \text{ is in } \mathcal{A}' \text{ and } \mathcal{B}|_{C''} \text{ is in } \mathcal{A}''.$$

Thus, we have to prove that $\mathcal{B}|_{C'}$ globally satisfies all the rules in $R_g{}'$ and locally satisfies all the rules in $R_\ell{}'$ and similarly with respect to $\mathcal{B}|_{C''}$.

Let $\langle \Delta, \varphi \rangle \in R_g{}'$. $\mathcal{B}|_{C'}$ is an interpretation structure over C'. $\mathcal{B}|_{C'}$ and $\mathcal{B}$ have the same carrier set and partial order. Let α' be an assignment over $\mathcal{B}|_{C'}$. Since α' is also an assignment over $\mathcal{B}$ and $\mathcal{B}$ globally satisfies $\langle \Delta, \varphi \rangle$,

$$\text{if } [\![\delta]\!]^{\alpha'}_{\mathcal{B}} = \top \text{ for every } \delta \in \Delta \text{ then } [\![\varphi]\!]^{\alpha'}_{\mathcal{B}} = \top.$$

By Lemma 3.1.26, if $[\![\delta]\!]^{\alpha'}_{\mathcal{B}|_{C'}} = \top$ for every $\delta \in \Delta$ then $[\![\varphi]\!]^{\alpha'}_{\mathcal{B}|_{C'}} = \top$. Hence, $\mathcal{B}|_{C'}$ globally satisfies $\langle \Delta, \varphi \rangle$.

Consider now $\langle \Delta, \varphi \rangle \in R_\ell{}'$. Let α' be assignment over $\mathcal{B}|_{C'}$ and b in the carrier set of $\mathcal{B}|_{C'}$. Since $\mathcal{B}$ locally satisfies $\langle \Delta, \varphi \rangle$,

$$\text{if } b \leq [\![\delta]\!]^{\alpha'}_{\mathcal{B}} \text{ for every } \delta \in \Delta \text{ then } b \leq [\![\varphi]\!]^{\alpha'}_{\mathcal{B}}.$$

By Lemma 3.1.26, if $b \leq [\![\delta]\!]^{\alpha'}_{\mathcal{B}|_{C'}}$ for every $\delta \in \Delta$ then $b \leq [\![\varphi]\!]^{\alpha'}_{\mathcal{B}|_{C'}}$. Hence, $\mathcal{B}|_{C'}$ locally satisfies $\langle \Delta, \varphi \rangle$. Since L' is full, $\mathcal{B}|_{C'}$ is in $\mathcal{A}'$. Similarly, we conclude that $\mathcal{B}|_{C''}$ is in $\mathcal{A}''$. Thus $\mathcal{B}$ is in the class of interpretation structures of $L' \cup L''$ and therefore $L' \cup L''$ is full. ◁

Lemma 3.3.17 *The logic system resulting from fibring has* verum *provided that at least one of the given logic systems has verum.*

Proof. Let L' and L'' be logic systems. Assume L' has *verum* and let $\varphi \in L(C)$ be such that $\vdash^g_{L'} \varphi$ and $[\![\varphi]\!]^{\alpha'}_{\mathcal{B}'} = \top$ for every $\mathcal{B}'$ in the class of interpretation structures of L' and assignment α' over $\mathcal{B}'$. Using Remark 2.3.5,

$$\vdash^g_{L' \cup L''} \varphi.$$

Consider $\mathcal{B}$ in the class of interpretation structures of $L' \cup L''$ and α assignment over $\mathcal{B}$. Then $\mathcal{B}|_{C'}$ is in the class of interpretation structures of L' and, by Lemma 3.1.26, $[\![\varphi]\!]^{\alpha}_{\mathcal{B}|_{C'}} = [\![\varphi]\!]^{\alpha}_{\mathcal{B}}$. Hence $[\![\varphi]\!]^{\alpha}_{\mathcal{B}} = \top$ and, since $\mathcal{B}$ and $\mathcal{B}|_{C'}$ have the same top, we conclude that $L' \cup L''$ has *verum*. ◁

We now use the results on preservation of implication and equivalence by fibring presented in Subsection 2.3.2 of Chapter 2 to establish sufficient conditions for fibring to preserve global completeness. Recall the notion of Hilbert calculus with implication and equivalence given in Definitions 2.3.15 and 2.3.27.

Theorem 3.3.18 *The fibring while sharing implication and equivalence of full logic systems with implication, equivalence and* verum *is globally complete.*

Proof. Let L' and L'' be full logic systems with implication, equivalence and *verum*. By Proposition 2.3.28, $L' \cup L''$ has implication and equivalence hence, by Proposition 2.3.30, $L' \cup L''$ has *MTC*. By Proposition 3.3.17, $L' \cup L''$ has *verum*. By Theorem 3.3.16, $L' \cup L''$ is full. Finally, by Theorem 3.3.15, $L' \cup L''$ is globally complete. ◁

Example 3.3.19 Consider the logic systems $L'_{\mathbf{S4}}$ and $L''_{\mathbf{B}}$ presented in Example 3.3.9. Let $\bar{L}'_{\mathbf{S4}}$ and $\bar{L}''_{\mathbf{B}}$ be the corresponding extensions obtained by adding to the respective signatures a binary connective $\Leftrightarrow$ as well as the appropriate axioms for equivalence, as was done for classical logic in Example 2.3.20; at the semantic level, $\bar{L}'_{\mathbf{S4}}$ and $\bar{L}''_{\mathbf{B}}$ contain the obvious extensions of the corresponding interpretation systems. Then, the new logic systems are full, have equivalence and *verum* and therefore they are both globally complete. Moreover, in the fibring $\bar{L}'_{\mathbf{S4}} \cup \bar{L}''_{\mathbf{B}}$ implication and equivalence are shared. Hence,

$$\bar{L}'_{\mathbf{S4}} \cup \bar{L}''_{\mathbf{B}}$$

is also globally complete, corresponding to modal logic **S5** over a signature including equivalence. $\triangledown$

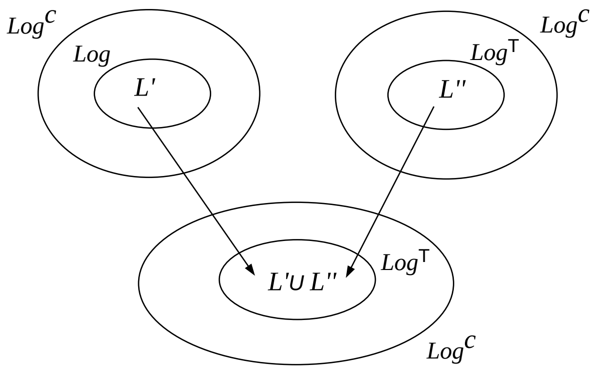


Figure 3.9: Preservation of completeness

The technique that was used to proving the preservation of completeness can be summarized in Figure 3.9. Let:

- *Log* be the class of full logic systems with the same implication and equivalence;
- $Log^{\top}$ be the subclass of *Log* composed by the logic systems that have *verum*;
- Log^c be the class of logic systems that are complete.

Let $L' \in Log$ and $L'' \in Log^{\top}$. Then their fibring $L' \cup L''$ is in $Log^{\top}$. We observe that:

- Theorem 3.3.15 establishes that $Log \in Log^c$;
- Theorem 3.3.16 shows that $L' \cup L'' \in Log$;
- Lemma 3.3.17 shows that $L' \cup L'' \in Log^{\top}$;
- Theorem 3.3.18 states that $L' \cup L'' \in Log^c$.

Hence, we prove that fibring of complete logic systems that are full, with implication, equivalence and *verum* is full, with implication, equivalence and *verum* and so is complete. That is, we do not prove in general that completeness is preserved. We prove that fullness is preserved and since fullness together with implication, equivalence and verum lead to completeness the fibring is complete.

3.4 Relationship with fibring by functions

The main objective of this section is to relate the fibring by functions with the algebraic fibring. Since the main difference appears to be on the definition of the models, we concentrate on semantic aspects. Moreover, it should be noted that the original version of fibring was introduced exclusively at the semantic level.

The fibring by functions as introduced in [104, 108] can be illustrated with a very simple example. Assume that you want to produce the fibring of two modal logics L' and L'' sharing the propositional symbols, the negation and the implication. Assume that L' has a (unary) space modality $\Box'$ and L'' with a (unary) time modality $\Box''$. Suppose also that both modal logics are endowed with a Kripke semantics.

Intuitively speaking a model of the fibring is a cloud of (two-dimensional) points in such a way that if we take a point we should be able to identify a Kripke structure for time and a Kripke structure for space. Hence assuming that W is the cloud of points, then we can consider two maps

$$f' : W \to KPK' \text{ and } f'' : W \to KPK''$$

assigning to each $w \in W$ a Kripke structure in KPK' (the class of Kripke structures for time) and a Kripke structure in KPK'' (the class of Kripke structures for space), respectively. Moreover, two different points in the clouds may involve different Kripke structures either in KPK' or KPK''. In Figure 3.10 we illustrate this situation with a concrete example.

We assume that we have models from m'_1 to m'_4 for space and models from m''_1 to m''_3 for time and have one model in the fibring (the one represented by the cloud). To point u_1 in the model of the fibring we associate model m'_1 for space and model m''_2 for time and to point u_2 we associate model m'_2 for space and model m''_3 for time.

In [240] a categorical account of fibring by functions is proposed using a fixed point construction that we will describe briefly. For an algebraic counterpart of this issue see also [34]. Before we have to introduce a point-based semantics.

We observe that if we are dealing with logics where the class of models is closed for unions (that is, the union of models is still a model) then we can replace the point based semantics by a one-to-one correspondence between the models of the fibring and the models of each component. In this case, the definition of morphism between interpretation systems is the same as in Remark 3.1.37.

In Figure 3.11, we assume that the logics for time and space are closed for unions. That is, the union m' of m'_1, m'_2, m'_3, m'_4 is still a model for space and the

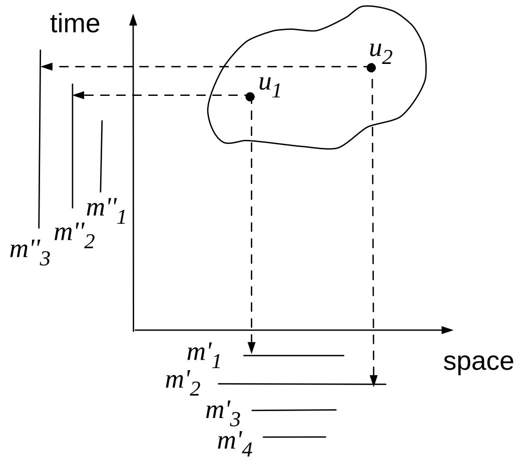


Figure 3.10: Model of fibring

union m'' of m''_1, m''_2, m''_3 is still a model for time. In this case, both u_1 and u_2 point to the same model of space m' and to the same model of time m''. We will see that it is always possible to close for unions any given logic.

From this point on, and adopting a general semantic perspective (in the modal sense), we see that getting rid of the set of points leads immediately to the algebraic approach. It is worthwhile to point out that even if the class of models of a logic is not closed for unions, we can close it without changing the semantic entailment.

We consider a signature C, the set of schema variables Ξ and language $L(C)$ as introduced in Section 2.1 of Chapter 2. We start by presenting the notions of structure over a signature, pre-interpretation system and interpretation system as introduced in [240]. Recall that we can also see $\wp U$ as the set of maps from the set U to $\{0, 1\}$ (the characteristic maps of the subsets of U). We use the notation Y^X to denote the set of all maps from set X to set Y.

Definition 3.4.1 Given a signature C, a C-*structure* is a pair $\langle U, \nu \rangle$ where U is a non-empty set and $\nu = \{\nu_k\}_{k \in \mathbb{N}}$ with $\nu_k : C_k \to (\wp U)^{((\wp U)^k)}$ for each $k \in \mathbb{N}$. ∇

The set U is called the set of *points*. We denote by $Str(C)$ the class of all structures over C. From the definition above, it is clear that $\langle \wp U, \nu \rangle$ is an algebra over C (recall the beginning of Section 3.1). Thus, structures over C are just powerset algebras over C with carrier set of the form $\wp U$ for a non-empty set U.

Definition 3.4.2 A *pre-interpretation system* is a triple $\langle C, M, A \rangle$ where C is a signature, M is a class and $A : M \to Str(C)$ is a map. ∇

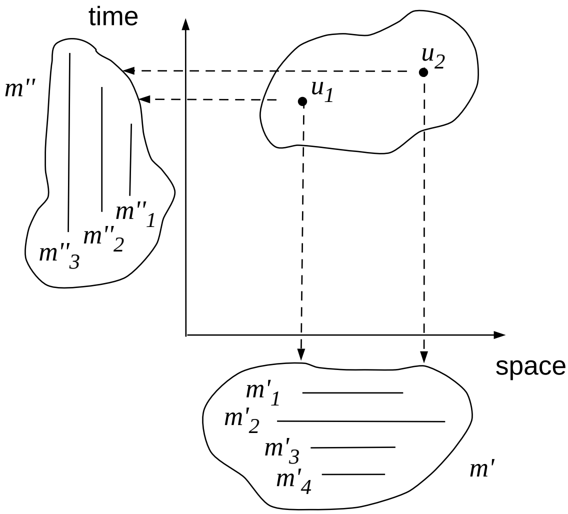


Figure 3.11: Model of fibring where the components are closed for union

The elements of M are the *models*. The map A associates a structure over C to each model. In the sequel, $\langle U_m, \nu_m \rangle$ denotes the structure $A(m)$ over C.

Definition 3.4.3 A pre-interpretation system $\langle C, M, A \rangle$ is an *interpretation system* provided that, for every $m_1 \in M$ and every bijection $f : U_{m_1} \to V$ there exists $m_2 \in M$ such that $U_{m_2} = V$ and

$$\nu_{m_2 k}(c)(b_1, \ldots, b_k)(f(u)) = \nu_{m_1 k}(c)(b_1 \circ f, \ldots, b_k \circ f)(u)$$

for every $k \in \mathbb{N}$, $c \in C_k$, $(b_1, \ldots, b_k) \in (\wp V)^k$ and $u \in U_{m_1}$. ∇

We use the notations $m_2 = f(m_1)$ and $\langle U_{m_2}, \nu_{m_2} \rangle = f(\langle U_{m_1}, \nu_{m_1} \rangle)$. Models m_1 and m_2 as above are called *equivalent*. Pre-interpretation systems can be easily enriched to obtain interpretation systems.

Prop/Definition 3.4.4 *The* enrichment *of a pre-interpretation system $\langle C, M, A \rangle$ is the interpretation system $\langle C, \overline{M}, \overline{A} \rangle$ where:*

- *$\overline{M}$ is the class of all pairs $\langle m, f \rangle$ where $m \in M$ and $f : U_m \to V$ is a bijection;*
- *$\overline{A}(\langle m, f \rangle) = f(A(m))$.* ∇

Example 3.4.5 An interpretation system corresponding to classical logic is $\langle C, M, A \rangle$ defined as follows:

- C is the classical signature in Example 2.1.2;
- M is the class of all pairs $\langle U, V\rangle$ where:
  - U is a singleton $\{u\}$;
  - $V : \mathbb{P} \to \{0, 1\}$.
- $A(\langle U, V\rangle) = \langle U, \nu\rangle$ where:
  - $\nu_0(p)(u) = V(p)$;
  - $\nu_1(\neg)(b)(u) = 1 - b(u)$;
  - $\nu_2(\Rightarrow)(b, b')(u) = 1$ if $b(u) \leq b'(u)$ and $\nu_2(\Rightarrow)(b, b')(u) = 0$ otherwise.

Here we identify, as usual, any set Y^{X^0} with Y. Clearly, if we substitute in the construction above the singleton $\{u\}$ by any non-empty set U, we obtain an interpretation system which is a particular case of the interpretation structures for classical logic presented in Example 3.1.3.

Of course, an analogous interpretation system for classical logic defined over the classical signature introduced in Example 2.1.4 can be defined, with minor adaptations. ∇

Example 3.4.6 An interpretation system for modal logic **K** is the triple $\langle C, M, A\rangle$ defined as follows:

- C is the modal signature in Example 2.1.4;
- M is the class of Kripke frames $\langle W, R\rangle$ defined as in Example 3.1.6;
- $A(\langle W, R\rangle) = \langle W, \nu\rangle$ where:
  - $\nu_1(\neg)(b)(w) = 1 - b(w)$;
  - $\nu_1(\Box)(b)(w) = \bigwedge_{w' : \langle w, w'\rangle \in R} b(w')$;
  - $\nu_2(\Rightarrow)(b, b')(w) = 1$ if $b(w) \leq b'(w)$ and $\nu_2(\Rightarrow)(b, b')(w) = 0$ otherwise.

If we consider Kripke structures and extend the mapping ν appropriately, we obtain an interpretation system corresponding to modal logic **K** defined over the modal signature of Example 2.1.2. ∇

In the example above, the symbol $\bigwedge$ denotes an infimum.

The definition of global and local entailments in (pre-)interpretation systems, to be introduced below, relies on the corresponding notions already defined.

Observe that a (pre-)interpretation system $I = \langle C, M, A\rangle$ induces an interpretation system $I_d = \langle C, \{\mathcal{B}_m : m \in M\}\rangle$ in the sense of Definition 3.1.16, where $\mathcal{B}_m$ is the interpretation structure

$$\mathcal{B}_m = \langle \wp U_m, \subseteq, \nu_m, U_m\rangle$$

for every $m \in M$. Using this, we define the following:

Definition 3.4.7 Consider the (pre-)interpretation system $I = \langle C, M, A\rangle$.
(i) The formula $\varphi \in L(C)$ is *globally entailed* by $\Gamma \subseteq L(C)$ in I, written $\Gamma \Vdash^g_I \varphi$, if $\Gamma \vDash^g_{I_d} \varphi$ in the sense of Definition 3.1.22.
(ii) The formula $\varphi \in L(C)$ is *locally entailed* by $\Gamma \subseteq L(C)$ in I, written $\Gamma \Vdash^l_I \varphi$, if $\Gamma \vDash^\ell_{I_d} \varphi$ in the sense of Definition 3.3.1. $\triangledown$

It is worth noting that

$$\Gamma \Vdash^l_I \varphi \text{ if and only if } (\bigcap_{\gamma\in\Gamma} [\![\gamma]\!]^\alpha_{\mathcal{B}_m}) \subseteq [\![\varphi]\!]^\alpha_{\mathcal{B}_m}$$

for every assignment α over the interpretation structure $\mathcal{B}_m$ and every $m \in M$.

As before, $\Gamma^{\Vdash^g_I}$ is the set of formulas globally entailed by Γ in I and $\Gamma^{\Vdash^l_I}$ is the set of formulas locally entailed by Γ in I. Obviously ${}^{\Vdash^g_I}$ and ${}^{\Vdash^l_I}$ are closure operators, because they coincide, respectively, with ${}^{\vDash^g_{I_d}}$ and ${}^{\vDash^\ell_{I_d}}$.

Note that these notions of global and local entailments are preserved by enrichment. In fact, considering the pre-interpretation system I and its enrichment $\overline{I}$, for every $\Gamma \subseteq L(C)$, the following conditions hold:

- $\Gamma^{\Vdash^g_I} = \Gamma^{\Vdash^g_{\overline{I}}}$, and
- $\Gamma^{\Vdash^l_I} = \Gamma^{\Vdash^l_{\overline{I}}}$.

Unconstrained fibring and constrained fibring of these interpretation systems are now described. When considering unconstrained fibring of two logics, no constructors are shared. Recall that, given two signatures C' and C'', their *disjoint union* is a signature $C' \oplus C'' = \{(C' \oplus C'')_k\}_{k\in\mathbb{N}}$ where $(C' \oplus C'')_k$ is a disjoint union of the sets C'_k and C''_k. In the sequel, $i' : C' \to C' \oplus C''$ denotes the family $\{i'_k\}_{k\in\mathbb{N}}$ of the natural injections $i'_k : C'_k \to (C' \oplus C'')_k$. Similarly with respect to $i'' : C'' \to C' \oplus C''$.

Prop/Definition 3.4.8 *Let $I' = \langle C', M', A'\rangle$ and $I'' = \langle C'', M'', A''\rangle$ be interpretation systems. Then, their* unconstrained fibring *(by functions) is the interpretation system*

$$I' \oplus I'' = \langle C, M, A\rangle$$

defined, using $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$, as follows:

- *C is $C' \oplus C''$*
- *M is the class of all pairs $m = \langle U, \tau\rangle$ such that:*
  - *U is a non-empty set;*
  - *$\tau = \{\tau_u\}_{u\in U}$ where each $\tau_u = \langle \tau'_u, \tau''_u\rangle$ such that $\tau'_u \in M'$, $\tau''_u \in M''$;*
  - *$U_{\tau'_u} = \{v \in U : \tau'_v = \tau'_u\}$, $U_{\tau''_u} = \{v \in U : \tau''_v = \tau''_u\}$ for each $u \in U$;*

– *letting* $O : \wp U \to \wp U$ *be such that*

$$O(V) = \bigcup_{v \in V} (U_{\tau'_v} \cup U_{\tau''_v})$$

then $\bigcup_{n \in \mathbb{N}} O^n(\{u\}) = U$ *for every* $u \in U$.

- $A(\langle U, \tau \rangle) = \langle U, \nu \rangle$ *where, for* $k \in \mathbb{N}$, $\vec{b} = (b_1, \dots, b_k) \in \wp U^k$ *and* $u \in U$:

$$\nu_k(i'_k(c'))(\vec{b})(u) = \begin{cases} \nu_{\tau'_u k}(c')(b_1 \circ inc'_u, \dots, b_k \circ inc'_u)(u) & \text{if } u \in U_{\tau'_u} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\nu_k(i''_k(c''))(\vec{b})(u) = \begin{cases} \nu_{\tau''_u k}(c'')(b_1 \circ inc''_u, \dots, b_k \circ inc''_u)(u) & \text{if } u \in U_{\tau''_u} \\ 0 & \text{otherwise} \end{cases}$$

being $inc'_u : U_{\tau'_u} \to U$ *and* $inc''_u : U_{\tau''_u} \to U$ *the corresponding inclusions.* ∇

Figure 3.12 illustrates the relationship between the carrier set of a model in the fibring with the carrier sets of the models in the components that are involved in that model. Let

$$U = \langle \{u_1, u_2, u_3, u_4, u_5, u_6\}, \tau \rangle$$

be the component of a model in the fibring $I' \oplus I''$ where:

- $\tau'_{u_1} = \tau'_{u_4} = \tau'_{u_5}$ and $\tau'_{u_2} = \tau'_{u_3} = \tau'_{u_6}$;
- $\tau''_{u_1} = \tau''_{u_2}$ and $\tau''_{u_3} = \tau''_{u_4} = \tau''_{u_5} = \tau''_{u_6}$.

Hence, for example, u_1, u_4, u_5 belong to the carrier set of the same model in I' whereas u_1, u_2 belong to the carrier set of the same model in I''. Moreover, all the points in the carrier sets of the component models are represented in the model of the fibring. The denotation of a connective of C' at say point u_1 in the fibring will be calculated in the same way as it is evaluated in point u'_1 in model τ'_{u_1} and the denotation of a connective of C'' at the same point u_1 in the fibring will be calculated in the same way as it is evaluated in point u''_1 in model τ''_{u_1}.

The following example illustrates the unconstrained fibring of interpretation systems in this setting.

Example 3.4.9 Consider the interpretation systems $I'_{\mathbf{S4}} = \langle C, M', A' \rangle$ and $I''_{\mathbf{B}} = \langle C, M'', A'' \rangle$, corresponding respectively to modal logics **S4** and **B**, where C is the modal signature in Example 2.1.2, M' is the class of Kripke structures with a

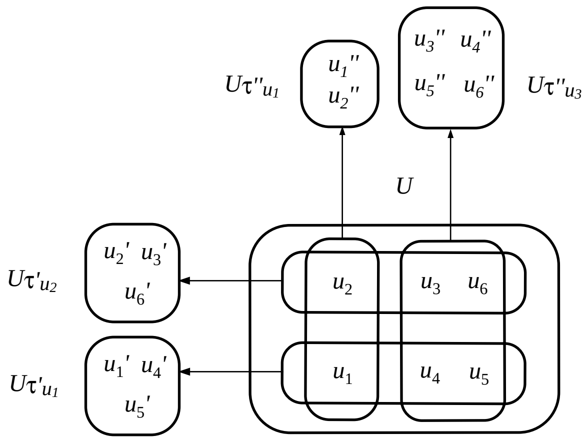


Figure 3.12: Carrier set in a model of the fibring by functions

reflexive and transitive accessibility relation, M'' is the class of Kripke structures with a symmetric accessibility relation and A', A'' are as in Example 3.4.6. Their unconstrained fibring by functions is

$$I'_{\mathbf{S4}} \oplus I''_{\mathbf{B}} = \langle C' \oplus C'', M, A\rangle$$

where, using $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$,

- M is the class of all fibred models $\langle U, \tau\rangle$ such that $\tau' = \{\tau'_u\}_{u\in U}$ (respectively $\tau'' = \{\tau''_u\}_{u\in U}$) induce reflexive and transitive (respectively symmetric) Kripke structures over U.

- $A(\langle U, \tau\rangle) = (U, \nu)$ with

  - $\nu_1(i'_1(\Box))(b)(u) = \bigwedge_{u':\langle u,u'\rangle\in R'_u} b \circ inc'(u')$ where $\tau'_u = \langle U_{\tau'_u}, R'_u, V'_u\rangle$;
  - $\nu_1(i''_1(\Box))(b)(u) = \bigwedge_{u'':\langle u,u''\rangle\in R''_u} b \circ inc''(u'')$ where $\tau''_u = \langle U_{\tau''_u}, R''_u, V''_u\rangle$.

In fact, τ' induces $\langle W', R', V'\rangle$ where

- $W' = U$;

- $R' = \bigcup_{u\in U} R'_u$;

- $V'(i'(p)) = \bigcup_{u\in U} V'_u(p)$ for each $p \in C'_0$.

The family τ'' induces $\langle W'', R'', V''\rangle$ in a similar way. ∇

When we want to share some constructors in the signatures C' and C'', we must consider constrained fibring. In Section 3.1, for simplicity, shared constructors were always the constructors in $C' \cap C''$. Herein, following [240], we are going to consider also the case where we share different symbols.

To this end, we need a signature C^{sh} and two families

$$f' = \{f'_k\}_{k\in\mathbb{N}} \text{ and } f'' = \{f''_k\}_{k\in\mathbb{N}}$$

of injective maps $f'_k : C^{sh}_k \to C'_k$ and $f''_k : C^{sh}_k \to C''_k$. The maps establish the constructors to be shared: for each $c \in C^{sh}_k$, $f'_k(c)$ and $f''_k(c)$ are shared. In categorical terms, two families f' and f'' as above are monomorphisms

$$f' : C^{sh} \to C' \text{ and } f'' : C^{sh} \to C'' \text{ in } \mathbf{Sig}.$$

Thus, we say that a diagram in **Sig** such that f' and f'' are monomorphisms in **Sig** is a *sharing diagram* between C' and C'' (see Figure 3.13).

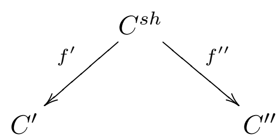


Figure 3.13: Sharing diagram

The constrained fibring (by functions) of two interpretation systems is built on top of their unconstrained fibring identifying shared symbols and their denotations and is a pushout in the relevant category.

Prop/Definition 3.4.10 *Let $I' = \langle C', M', A'\rangle$ and $I'' = \langle C'', M'', A''\rangle$ be interpretation systems and let C^{sh}, $f' : C^{sh} \to C'$ and $f'' : C^{sh} \to C''$ inducing a sharing between C' and C''. Then, the* constrained fibring by functions *of I' and I'' is the interpretation system*

$$I' \overset{f'f''}{\oplus} I'' = \langle C, M, A\rangle$$

where, considering the unconstrained fibring $I' \oplus I'' = \langle C' \oplus C'', M' \oplus M'', A' \oplus A''\rangle$, using $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$ such that $C^{sh} \cap C' \oplus C'' = \emptyset$ and letting $A' \oplus A''(m) = \langle U_m, \nu_m\rangle$,

- *$C = \{C_k\}_{k\in\mathbb{N}}$ is such that $C_k = C^{sh}_k \cup i'_k(C'_k \backslash f'_k(C^{sh}_k)) \cup i''_k(C''_k \backslash f''_k(C^{sh}_k))$;*
- *M is the subclass of all elements m of $M' \oplus M''$ such that:*
  - *$\nu_{mk}(c_1) = \nu_{mk}(c_2)$ for each $k \in \mathbb{N}$ and $c_1, c_2 \in C_k$ such that $i'_k \circ f'_k(c) = c_1$ and $i''_k \circ f''_k(c) = c_2$ for some $c \in C^{sh}_k$;*
- *$A(m) = \langle U_m, \nu'\rangle$ where*

– $\nu'_k(c) = \nu_{mk}(c)$ *for each* $k \in \mathbb{N}$ *and* $c \in i'_k(C'_k \backslash f'_k(C^{sh}_k)) \cup i''_k(C''_k \backslash f''_k(C^{sh}_k))$;

– $\nu'_k(c) = \nu_{mk}(i'_k \circ f'_k(c))$ *for each* $k \in \mathbb{N}$ *and* $c \in C^{sh}_k$. ▽

Example 3.4.11 Recall the unconstrained fibring of $I'_{\mathbf{S4}}$ and $I''_{\mathbf{B}}$ presented in Example 3.4.9. If we share the propositional constructors $\mathbb{P}$, $\neg$ and $\Rightarrow$ and consider $i'(\Box) = \Box'$ and $i''(\Box) = \Box''$, the constrained fibring of $I'_{\mathbf{S4}}$ and $I''_{\mathbf{B}}$ corresponds to a bimodal logic.

The Kripke structure associated to each fibred models $\langle U, \tau\rangle$ is now

$$\langle W, R', R'', V\rangle$$

where $W = W' = W'' = U$, R' is a reflexive and transitive relation, R'' is a symmetric relation and $V(p) = V'(i'(p)) = V''(i''(p))$ for each $p \in \mathbb{P}$. Observe that, since the two modalities are inequivalent, this interpretation system does not correspond to modal logic **S5**. ▽

When we endow Hilbert systems with this kind of interpretation systems we get a new class of logic systems, and then notions of soundness and completeness can be considered. In [240], a result of soundness preservation is presented.

In order to understand the relationship between the point based approach and the algebraic approach to semantics, it is better to introduce a general point based semantics (the terminology is borrowed from general frames in modal logic [265, 24]). The general point based semantics was essential to achieve completeness preservation as in [282]. We now present the corresponding structures and interpretation systems, called *general structures* and *general interpretation systems*.

Definition 3.4.12 A *general structure* over C is a triple $\langle U, \mathcal{B}, \nu\rangle$ where U is a non-empty set, $\mathcal{B}$ is a non-empty subset of $\wp U$, and $\nu = \{\nu_k\}_{k\in\mathbb{N}}$ is a family of functions such that $\nu_k : C_k \to \mathcal{B}^{(\mathcal{B}^k)}$ ▽

The class of all general structures over C is denoted by $gStr(C)$. As above, U is the set of *points*. The set $\mathcal{B}$ is the set of *admissible valuations*. The particular case where $\mathcal{B}$ is $\wp U$ corresponds to the structures over C presented above.

In the sequel, given a map g with domain S and $X = \langle X_1, \ldots, X_k\rangle \in (\wp S)^k$, for simplicity, we use $g(X)$ for $\langle g(X_1), \ldots, g(X_k)\rangle$. Similarly,

$$X \cap Y^k$$

abbreviates $\langle X_1 \cap Y, \ldots, X_k \cap Y\rangle$.

Two structures over C $\langle U, \mathcal{B}, \nu\rangle$ and $\langle U', \mathcal{B}', \nu'\rangle$ are said to be *isomorphic* whenever there is a bijection $f : U \to U'$ such that, for every $k \in \mathbb{N}$, $c \in C_k$ and $b \in \mathcal{B}^k$, $\nu_k(c)(b) = \nu'_k(c)(f(b))$ and $\mathcal{B}' = \{f(b) : b \in \mathcal{B}\}$.

Definition 3.4.13 A *general pre-interpretation system* is a triple $\langle C, M, A\rangle$ in which M is a class and A is a map from M into $gStr(C)$. ▽

As before, the elements of M are *models* and $\langle U_m, \mathcal{B}_m, \nu_m\rangle$ denotes $A(m)$. General interpretation systems are general pre-interpretation systems closed under isomorphism and disjoint union of structures over C .

Prop/Definition 3.4.14 *A general pre-interpretation system* $\langle C, M, A\rangle$ *is a* general interpretation system *if it is closed under isomorphic images and disjoint unions, that is,*

- *if* $\langle U, \mathcal{B}, \nu\rangle$ *and* $\langle U', \mathcal{B}', \nu'\rangle$ *are isomorphic and* $\langle U, \mathcal{B}, \nu\rangle = A(m)$ *for some* $m \in M$*, then there exists* $m' \in M$ *such that* $A(m') = \langle U', \mathcal{B}', \nu'\rangle$*;*
- *if* $U_n \cap U_{n'} = \emptyset$ *for all* $n \neq n'$ *in some* $N \subseteq M$*, then there exists* $m \in M$ *such that:*
  - $U_m = \bigcup_{n\in N} U_n$*;*
  - $\mathcal{B}_m = \{b \in \wp U_m : b \cap U_n \in \mathcal{B}_n$ *for all* $n \in N\}$
  - *for every* $k \in \mathbb{N}$ *and* $b \in \mathcal{B}_m^k$, $\nu_{mk}(c)(b) = \bigcup_{n\in N} \nu_{nk}(c)(b \cap U_n^k)$. ▽

Given any general pre-interpretation system $\langle C, M, A\rangle$, we can always obtain the smallest interpretation system $\langle C, M, A\rangle^c$ containing it, by making it closed under isomorphic images and disjoint unions in the obvious way.

It is also important to consider the closure under subalgebras of an interpretation systems. Given a structure $\langle U, \mathcal{B}, \nu\rangle$ over C, we say that $\mathcal{B}'$ is a *ν-subalgebra* of $\mathcal{B}$ whenever $\mathcal{B}' \subseteq \mathcal{B}$ and $\mathcal{B}'$ is closed under the operations $\nu_k(c)$ for all $k \in \mathbb{N}$ and $c \in C_k$. An interpretation system $\langle C, M, A\rangle$ is said to be *closed under subalgebras* whenever, for every $m \in M$ and every ν_m-subalgebra $\mathcal{B}'$ of $\mathcal{B}_m$, there is a model $m' \in M$ such that $U_{m'} = U_m$, $\mathcal{B}_{m'} = \mathcal{B}'$, and, for all $k \in \mathbb{N}$ and $c \in C_k$, $\nu_{m'}(c) = \nu_m(c)|_{\mathcal{B}'^k}$.

Denotations of formulas and global and local entailments are defined as expect. An *assignment* into a model $m \in M$ of a general (pre-)interpretation system $\langle C, M, A\rangle$ is map $\alpha : \Xi \to \mathcal{B}_m$, and the *denotation* map

$$[\![.]\!]_m^\alpha : L(C) \to \mathcal{B}_m$$

is inductively defined as expected using α and the map ν_m. Note that assignments to schema variables and denotation of formulas belong to $\mathcal{B}_m$, that is, they are admissible valuations. Global entailment and local entailment are as in Definition 3.4.7. Again, the global entailment operator induced by a general pre-interpretation system $\langle C, M, A\rangle$ and the one induced by its closure $\langle C, M, A\rangle^c$ are the same. Similarly with respect to local entailment. Similar remarks also hold when considering closure under subalgebras. In the sequel, all interpretation are assumed to be closed under subalgebras.

Finally, we describe the fibring mechanism for general interpretation structures. We start with unconstrained fibring.

Prop/Definition 3.4.15 *Let $gI' = \langle C', M', A'\rangle$ and $gI'' = \langle C'', M'', A''\rangle$ be general interpretation systems. Then, their* unconstrained fibring by functions *is the general interpretation system*

$$gI' \oplus gI'' = \langle C, M, A\rangle$$

defined, using $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$, as follows:

- *C is $C' \oplus C''$*
- *M is the subclass of $M' \times M''$ composed of the pairs $\langle m', m''\rangle$ such that:*
  - *$U_{m'} = U_{m''}$ and $\mathcal{B}_{m'} = \mathcal{B}_{m''}$;*
- *$A(\langle m', m''\rangle) = \langle U, \mathcal{B}, \nu\rangle$ where:*
  - *$U = U_{m'} (= U_{m''})$; $\mathcal{B} = \mathcal{B}_{m'} (= \mathcal{B}_{m''})$;*
  - *$\nu_k(i'(c')) = \nu_{m'k}(c')$ for each $c' \in C'_k$;*
  - *$\nu_k(i''(c'')) = \nu_{m''k}(c'')$ for each $c'' \in C''_k$.* ∇

The fibred models are just pairs of models, one from each original general interpretation system, with equal set of points and set of admissible valuations. The structure over C associated to each fibred model includes their common set of points and set of admissible valuations, and each constructor is interpreted as in the interpretation system it comes from.

Herein, we take profit of the fact that interpretation systems are closed under disjoint unions. Indeed, to each model m of the fibring we assign just one model of each component, say m' and m''. The model m' is the disjoint union of the models that were assigned to m. The model m'' is obtained in a similar way.

Next, we consider constrained fibring. As above, the constrained fibring of two general interpretation systems is built on top of their unconstrained fibring identifying shared symbols and their denotations.

Prop/Definition 3.4.16 *Let $gI' = \langle C', M', A'\rangle$ and $gI'' = \langle C'', M'', A''\rangle$ be general interpretation systems and let*

$$\{f' : C^{sh} \to C',\ f'' : C^{sh} \to C''\}$$

be a sharing diagram between C' and C''. Then, the constrained fibring by functions *of gI' and gI'' is the interpretation system*

$$gI' \overset{f'f''}{\oplus} gI'' = \langle C, M, A\rangle$$

where, considering the unconstrained fibring $gI' \oplus gI'' = \langle C' \oplus C'', M' \oplus M'', A' \oplus A''\rangle$, using $i' : C' \to C' \oplus C''$ and $i'' : C'' \to C' \oplus C''$ such that $C^{sh} \cap C' \oplus C'' = \emptyset$, and letting $A' \oplus A''(m) = \langle U_m, \mathcal{B}_m, \nu_m\rangle$,

- *C is defined as in Prop/Definition 3.4.10;*
- *M is the subclass of all elements m of $M' \oplus M''$ satisfying the same requirements as in Prop/Definition 3.4.10;*
- *$A(m) = \langle U_m, \mathcal{B}_m, \nu' \rangle$ where ν' is defined as in Prop/Definition 3.4.10.* ∇

Unconstrained fibring by functions of interpretation systems is a coproduct, and constrained fibring by functions is a pushout. Logic systems including general interpretation systems are defined as expected, as well as the usual notions of soundness and completeness. In [282], results of soundness preservation and completeness preservation in this setting are presented.

Observe that a general structure $\langle U, \mathcal{B}, \nu \rangle$ over C is a power-set algebraic structure. That is, the set of truth values is a subset of the power set of U. The role of U is also important for stating the truthfulness of a formula. Hence, we can abstract from this structure in an easy way by having a set of truth values with a top element as we do when defining algebraic structures in Section 3.1.

3.5 Final remarks

In this chapter, algebraic fibring was introduced. The basic semantic structure is an ordered algebra whose carrier set represents the set of truth values and the operations correspond to the denotations of the connectives. Moreover, the order relation allows to compare truth values in each structure. The top element of each algebra corresponds to theorems.

Algebraic fibring is a step towards being able to define fibring, at the semantic level, of logics that do not have the same kind of semantics. We assume that we are able to generate the ordered algebras from the semantic structures of the logic with no loss. That is, the semantic entailment of the logic is captured by the ordered algebras. For example, if we consider modal logic endowed with Kripke semantics we can extract from each Kripke structure an ordered algebra by considering the set of all subsets of the set of worlds as truth values, the set of worlds as the top and by comparing truth values by means of inclusion of sets. Observe that this is the process used to generate a modal algebra from a Kripke structure.

Since fibring by functions seems to be very different, we discussed how algebraic fibring captures fibring by functions. In a nutshell we can say that algebraic fibring is related to fibring by functions as modal algebras are related with Kripke structures. A bit of history is in order at this point. The first step when one is introduced to modal logic is to go through Kripke structures. Then we learn that not every modal logic is complete with respect to Kripke semantics and so we learn about general Kripke structures. When we have the set of valuations, in our terminology the set of truth values, modal algebras are easily understood as an abstraction of the point-based semantics.

The fibring by functions is very intuitive even in a geometric way. In the end, fibring of point-based models corresponds to a fixed point construction. The main drawback of the definition is that to a model of the fibring we can assign a set of models of each original logic. This can be simplified if we assume that our logic is closed for unions. In this situation, to each model m of the fibring we assign just one model of each component, say m' and m''. The model m' is the union of the models that were previously assigned to m. Similarly with respect to the model m''. As it is well known, many logics are not closed for unions. For instance, modal logic, in general, is closed for unions but linear temporal logic is not. However, it is possible to close for unions any logic preserving the semantic entailment (for more details on this issue see [282]). When we are concerned with completeness in the presence of the point-based semantics, immediately general structures are in order. And from there to algebraic semantics is just a small abstraction step.

Concerning preservation properties, in this chapter it is worthwhile to make some a posteriori comments. The first one is that in many cases we can only prove sufficient conditions for preservation, as in completeness. The other is a strategy that can be important to prove other metalogical properties. When we want to prove preservation of a property, we should look for sufficient conditions for a logic to have that property and then prove the preservation of the sufficient conditions. For instance, among others, fullness is a requirement for a logic to be complete. Then we show that fullness is preserved by fibring.

Another important aspect to report is that congruence is not always preserved by fibring. Basically, congruence is a metaproperty that cannot be expressed in the object logic. Please see the difference, for instance, with the metatheorem of deduction in Chapter 2. An open problem is to show that congruence is preserved by logics whose rules are schematic. That is, all the counterexamples found related to non-preservation of congruence were given for logics (strange ones) where rules are not schematic. Of course, for solving this problem one should find a good definition of schematic rule.

Another open problem is to see whether the relaxation of the metatheorem of deduction in the lines of Chapter 2 can provide sufficient conditions for the preservation of completeness of more logics.

We believe that the people interested in category theory could provide an alternative version of the adopted algebraic structure in categorical terms. This would provide a more abstract structure covering also logics that have a categorical semantics from the very beginning. Some steps in this direction will be presented in Chapter 7.

Finally, we believe that preservation of more properties should be further explored. Among them the preservation of interpolation using semantic techniques like the amalgamation property is also worth investigating. The results obtained in [190, 191, 192, 193] should be taken as an inspiration both for local and global entailments. Preservation of weak completeness should also be considered taking into account the work in [98] for temporal logics and in [92] for non-normal modal logics, with the aim of relaxing the assumptions and covering more logics.

Chapter 4

Heterogeneous fibring

In Chapter 2, we have studied the fibring of logics in a homogeneous scenario: both logics were presented in the same way, through Hilbert calculi. However, this is not usually the case. Often, we have an heterogeneous setting, that is, we are given two logics presented in different ways. We may have, for instance, two modal logics such that one is presented by a Hilbert calculus and the other is presented by a sequent calculus or even semantically presented by Kripke structures.

In this chapter we study the fibring of logics in this heterogeneous setting, in particular, the fibring of logics when:

(i) one is presented in a semantic way and the other is presented by a calculus;

(ii) both are presented by calculi but these are of different nature;

(iii) both are presented by different semantic structures.

This latter case was tackled in Chapter 3 where we provide an algebraic structure where a large number of semantic structures can be accommodated. In this chapter we concentrate on examples of fibring of Hilbert, sequent and tableau calculi. From the semantic point of view we investigate fibring of logics presented by satisfaction systems.

The solution to (i), which is also a solution to (ii), is to define fibring of consequence systems. The motivation is very easy: a logic presented either in a syntactic or in a semantic way always induce at least a consequence operator. For instance in Chapter 2, an Hilbert calculus with global and local rules induces two consequence operators: a global and a local one. Since the rules are related so are the consequence operators.

The idea is to get in a first step a consequence system for each of the given logics and then consider their fibring. However, using this approach, the constructive nature of derivations is lost. Hence, this solution is not a satisfactory one when dealing with logics presented by calculus because no trace is kept of the derivations that may exist in the original calculi. For this reason, we provide another solution to (ii) introducing abstract proof systems and their fibring. An abstract proof systems intends to abstract the essential properties of logics presented syntactically via some notion of derivation. The fibring of proof systems keeps the constructive nature of derivations. Hilbert, sequent and tableau calculi all induce proofs systems. Hence, in this approach, we first get from each calculus the corresponding proof system and then we consider their fibring.

In both solutions for heterogeneous fibring we also use the ghost technique that was introduced in Chapter 2 for investigating preservation of interpolation. In a nutshell we must be able to translate formulas from the fibring to the components.

Section 4.1 concentrates on consequence systems. To begin with we introduce sequent calculi and tableau calculi and show how they induce consequence systems. We also introduce satisfaction systems, generalizing the interpretation systems presented in Chapter 3, and show how they also induce consequence systems. Finally, we define fibring of consequence systems using a fixed point operator. Section 4.2 focuses on the notion of abstract proof system. We introduce the proof systems induced by Hilbert, sequent and tableau calculi and then we define fibring of proof systems. In both settings some preservation results are proved. We also discuss some relationships between consequence systems and proof systems. In Section 4.3, we present some final remarks.

The relevant material for this Chapter is the work presented in [68]. We refrain here of considering some preservation results that are there namely related to strong and weak semi-decidability. The reason is that in order to present those results we need some computability results on Gödelization of universes.

4.1 Fibring consequence systems

In this section we concentrate on consequence systems as a possible solution to heterogeneous fibring. We begin by presenting sequent calculi, tableau calculi and their corresponding consequence systems. From the semantic point of view, we consider satisfaction systems which, as expected, also induce consequence systems. Next, we define fibring of consequence systems and study some of its properties. Heterogeneous fibring of logics is then achieved in two steps: we first get the consequence system induced by each logic and then we consider their fibring.

4.1.1 Induced consequence systems

Herein we show how sequent calculi and tableau calculi induce consequence systems. We also introduce satisfaction systems and their corresponding consequence systems.

Recall from Chapter 1 that a consequence system is a pair $\langle C, \mathsf{C}\rangle$ where C is a signature and $\mathsf{C} : \wp L(C) \to \wp L(C)$ is a map with the following properties:

(i) $\Gamma \subseteq \mathsf{C}(\Gamma)$ — extensivity

(ii) if $\Gamma_1 \subseteq \Gamma_2$ then $\mathsf{C}(\Gamma_1) \subseteq \mathsf{C}(\Gamma_2)$ — monotonicity

(iii) $\mathsf{C}(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}(\Gamma)$ idempotence.

Recall also that a consequence system is said to be *compact* if

$$\mathsf{C}(\Gamma) = \bigcup_{\Phi \in \wp_{fin} \Gamma} \mathsf{C}(\Phi)$$

for each $\Gamma \subseteq L(C)$.

Herein, we also consider the notion of *quasi-consequence system*, that is, a pair $\langle C, \mathsf{C} \rangle$ verifying conditions (i) and (ii) above. Quasi-consequence systems can be used to express properties of one-step consequence.

A consequence system is said to be *closed for renaming substitutions* if

$$\sigma(\mathsf{C}(\Gamma)) \subseteq \mathsf{C}'(\sigma(\Gamma))$$

for every $\Gamma \subseteq L(C)$ and every renaming substitution σ (that is, $\sigma(\xi) \in \Xi$ for each $\xi \in \Xi$) and it is said to be *structural* if that inclusion holds for every substitution.

Throughout this chapter we work with propositional based signatures where each set of connectives C_k is denumerable.

Sequent calculi

Sequent calculi are also often called Gentzen calculi (the mathematician Gerhard Gentzen was the one that introduced this kind of calculi see [124]). A sequent calculi includes a signature and a set of sequent inference rules.

Definition 4.1.1 A *sequent* over a signature C is a pair $\langle \Delta_1, \Delta_2 \rangle$ where Δ_1 and Δ_2 are multi-sets of formulas in $L(C)$. A *sequent inference rule* is a pair $\langle \{\Theta_1, \ldots, \Theta_n\}, \Theta \rangle$ where $\Theta_1, \ldots, \Theta_n$ and Θ are sequents. $\triangledown$

Some variants of sequents are possible namely when Δ_1 and Δ_2 are either sets or sequences. In our case we work with multi-sets that is we have no order and can have more than one occurrence of the same element.

Sequents are often denoted by $\Delta_1 \to \Delta_2$. The multi-set Δ_1 is said to be the *antecedent* of the sequent and the multi-set Δ_2 its *consequent*. If $\Delta_1 \cap \Delta_2 \neq \emptyset$, the sequent is said to be an *axiom*.

A sequent inference rule may also be represented by

$$\frac{\Theta_1 \quad \ldots \quad \Theta_n}{\Theta}$$

Each sequent Θ_i is said to be a *premise* of the sequent inference rule and Θ its *conclusion*. Some sequent inference rules are particularly relevant and are given special names. The following inference rules are often considered.

- Structural rules:

$$\frac{\xi_1, \Delta_1 \to \Delta_2 \quad \Delta_1 \to \Delta_2, \xi_1}{\Delta_1 \to \Delta_2} \ \mathsf{Cut}$$

$$\frac{\Delta_1 \to \Delta_2}{\Delta_1 \to \Delta_2, \xi_1} \ \mathsf{RW} \qquad \frac{\Delta_1 \to \Delta_2}{\xi_1, \Delta_1 \to \Delta_2} \ \mathsf{LW}$$

$$\frac{\Delta_1 \to \xi_1, \xi_1, \Delta_2}{\Delta_1 \to \xi_1, \Delta_2} \ \mathsf{RC} \qquad \frac{\Delta_1, \xi_1, \xi_1 \to \Delta_2}{\Delta_1, \xi_1 \to \Delta_2} \ \mathsf{LC}$$

  where Δ_1 and Δ_2 are multi-sets of formulas in $L(C)$. The labels RW, LW, RC and LC respectively stand for right weakening, left weakening, right contraction and left contraction.

- Left rule for connective c: sequent inference rule whose conclusion is

$$(c(\varphi_1, \ldots, \varphi_n)), \Delta_1 \to \Delta_2$$

- Right rule for connective c: sequent inference rule whose conclusion is

$$\Delta_1 \to \Delta_2, (c(\varphi_1, \ldots, \varphi_n))$$

As a simplification we are not assuming that the connective rules have provisos like for example this element should be fresh in some set. The reason can be easily explained. We are not presenting a general theory of sequents but only using them as examples for heterogeneous fibring.

We now introduce the notion of sequent calculus.

Definition 4.1.2 A *sequent calculus* is a pair $G = \langle C, R \rangle$, where C is a signature and R is a set of sequent inference rules. $\triangledown$

In a sequent calculus, the set of sequent inference rules usually includes specific left and right rules for the connectives and in many cases also structural rules. We are also excluding from our examples labeled sequent calculi like for example those that are used for modal logic and for finite-valued logics.

Given a sequent calculus G, sequents can be derivable from a set of sequents, using the sequent inference rules.

Definition 4.1.3 A sequent s is *derivable* from a set $\mathcal{H}$ of sequents in G, denoted by

$$\mathcal{H} \vdash_G s$$

if there is a finite sequence $\Delta_{1,1} \to \Delta_{2,1} \ldots \Delta_{1,n} \to \Delta_{2,n}$ of sequents such that:

- $\Delta_{1,1} \to \Delta_{2,1}$ is s;
- for each $i = 1, \ldots, n$, one of the following holds:

- $\Delta_{1,i} \to \Delta_{2,i}$ is an axiom sequent;
- $\Delta_{1,i} \to \Delta_{2,i} \in \mathcal{H}$;
- there exist a sequent inference rule $\langle\{\Theta_1,\ldots,\Theta_k\},\Omega\rangle$ in G and a substitution σ such that $\sigma(\Theta_j) \in \{\Delta_{1,i+1} \to \Delta_{2,i+1},\ldots \Delta_{1,n} \to \Delta_{2,n}\}$, for $j = 1,\ldots,k$, and $\Delta_{1,i} \to \Delta_{2,i} = \sigma(\Omega)$.

The sequence $\Delta_{1,1} \to \Delta_{2,1} \ldots \Delta_{1,n} \to \Delta_{2,n}$ is a *derivation* of s from $\mathcal{H}$ em G. ▽

We say that a formula φ is *derivable* from the set of formulas Γ in G, indicated by

$$\Gamma \vdash_G \varphi$$

if $\vdash_G \Delta \to \varphi$ where Δ is a finite multi-set of formulas in Γ. As usual, $\vdash_G \varphi$ is used whenever Γ is an empty set. We denote by $\Gamma^{\vdash_G}$ the set of formulas *derivable from* Γ in G. When presenting a derivation, we add a justification for each sequent in the sequence: Ax when it is an axiom sequent, Hyp when it is a sequent in $\mathcal{H}$, and the name of the inference rule together with the corresponding positions of the instances of the premises in the other cases.

Example 4.1.4 A sequent calculus $G_{\mathbf{S4}}$ for modal logic **S4** includes the expected signature, the Cut rule and the following sequent inference rules [263], where $(\Box\Delta)$ and $(\Diamond\Delta)$ respectively stand for $\{(\Box\delta) : \delta \in \Delta\}$ and $\{(\Diamond\delta) : \delta \in \Delta\}$:

$$\frac{\Delta_1,\xi_1 \to \xi_2,\Delta_2}{\Delta_1 \to (\xi_1 \Rightarrow \xi_2),\Delta_2}\ R\Rightarrow \qquad \frac{\Delta_1 \to \xi_1,\Delta_2 \quad \Delta_1,\xi_2 \to \Delta_2}{\Delta_1,(\xi_1 \Rightarrow \xi_2) \to \Delta_2}\ L\Rightarrow$$

$$\frac{\Delta_1,\xi_1 \to \Delta_2}{\Delta_1 \to (\neg\,\xi_1),\Delta_2}\ R\neg \qquad \frac{\Delta_1 \to \xi_1,\Delta_2}{\Delta_1,(\neg\,\xi_1) \to \Delta_2}\ L\neg$$

$$\frac{(\Box\Delta_1) \to \xi_1,(\Diamond\Delta_2)}{\Delta_1',(\Box\Delta_1) \to (\Box\xi_1),(\Diamond\Delta_2),\Delta_2'}\ R\Box \qquad \frac{\Delta_1,\xi_1,(\Box\xi_1) \to \Delta_2}{\Delta_1,(\Box\xi_1) \to \Delta_2}\ L\Box$$

In the rules above, Δ_1, Δ_2, Δ_1' and Δ_2' are multi-sets of formulas in $L(C)$.

Note that we can extract a sequent calculus for propositional logic by eliminating $R\Box$ and $L\Box$. Observe also that weakening and contraction can be derived from these inference rules. The right and left rules for $\Diamond$ are easily obtained using the corresponding abbreviation.

As an example, we present below a derivation of $\to ((\Box\xi_1) \Rightarrow (\neg(\Box(\neg\xi_1))))$ from the empty set of hypothesis. To better understand the derivation, it is preferable to read it from the bottom to the top.

1	$\to ((\Box\xi_1) \Rightarrow (\neg(\Box(\neg\xi_1))))$	$R\Rightarrow$,2
2	$(\Box\xi_1) \to (\neg(\Box(\neg\xi_1)))$	$R\neg$,3
3	$(\Box\xi_1),(\Box(\neg\xi_1)) \to$	$L\Box$, 4
4	$(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)) \to$	$L\Box$,5
5	$(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)),(\neg\xi_1) \to$	$L\neg$,6
6	$(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)) \to \xi_1$	Ax

Hence, $\vdash_{G_{\mathbf{S4}}} \ \to ((\Box\xi_1) \Rightarrow (\neg(\Box(\neg\xi_1))))$ and therefore we can also conclude that $\vdash_{G_{\mathbf{S4}}} ((\Box\xi_1) \Rightarrow (\neg(\Box(\neg\xi_1))))$, or, using the usual abbreviation,

$$\vdash_{G_{\mathbf{S4}}} ((\Box\xi_1) \Rightarrow (\Diamond\xi_1))$$

meaning that **S4** includes the seriality axiom as a theorem. Observe that, in general, derivations in sequent calculi can be represented by trees. However, in order to study fibring of such systems it is more convenient to present derivations as sequences. For instance, the derivation above corresponds to the following tree.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)) \to \xi_1}{(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)),(\neg\xi_1) \to}\;L\neg}{(\Box\xi_1),\xi_1,(\Box(\neg\xi_1)) \to}\;L\Box}{(\Box\xi_1),(\Box(\neg\xi_1)) \to}\;L\Box}{(\Box\xi_1) \to (\neg(\Box(\neg\xi_1)))}\;R\neg}{\to ((\Box\xi_1) \Rightarrow (\neg(\Box(\neg\xi_1))))}\;R\Rightarrow
$$

∇

As we saw in Proposition 2.2.11, each Hilbert calculus induces a consequence system. The same holds for sequent calculi.

Proposition 4.1.5 *A sequent calculus $G = \langle C, R\rangle$ induces a compact and structural consequence system $\mathcal{C}(G) = \langle C, \vdash_G\rangle$.*

Tableau calculi

We now introduce tableau calculi. Tableau calculi are also called Smullyan calculi (as introduced by the logician Raymond Smullyan see [251, 250]). Most of the tableau calculi rely on the existence of a negation in the logic at hand. To be able to deal with as many logics as possible we avoid this assumption by considering tableau calculi over labeled formulas. Herein we only consider a very simple case for the labels. A *labeled formula* is a pair $\langle\varphi, i\rangle$, denoted by $i\!:\!\varphi$, where i is either 0 or 1. Intuitively speaking, $1\!:\!\varphi$ states that we want φ to be true and $0\!:\!\varphi$ means that we want φ to be false. We denote by $L^\lambda(C)$ the set of pairs $i\!:\!\varphi$ such that $i = 0, 1$ and $\varphi \in L(C)$.

Definition 4.1.6 A *tableau inference rule* is a pair $\langle\Upsilon, \mu\rangle$ where $\Upsilon \in \wp_{\text{fin}}\wp_{\text{fin}}L^\lambda(C)$ and $\mu \in L^\lambda(C)$. ∇

Given a tableau inference rule $\langle\Upsilon, \mu\rangle$, μ is the *conclusion* and each set in Υ is said to be an *alternative*. We can look at Υ as alternatives to μ. Given the alternatives $\Psi_1, \ldots, \Psi_n$, the tableau inference rule can also be represented by

$$\frac{\Psi_1 \ \ldots \ \Psi_n}{\mu}$$

We now present the notion of tableau calculus.

Definition 4.1.7 A *tableau calculus* is a pair $S = \langle C, R\rangle$ where C is a signature and R is a set of tableau inference rules. ▿

Tableau calculi usual include the excluded middle rule and positive and negative rules for the connectives, where,

- Excluded middle:

$$\frac{\{i\!:\!\xi_1, 1\!:\!\xi_2\} \quad \{i\!:\!\xi_1, 0\!:\!\xi_2\}}{i\!:\!\xi_1}\ EM$$

- Positive rule for connective c: tableau inference rule whose conclusion is

$$1\!:\!(c(\varphi_1,\dots,\varphi_n))$$

- Negative rule for connective c: tableau inference rule whose conclusion is

$$0\!:\!(c(\varphi_1,\dots,\varphi_n))$$

Given a tableau calculus S, sets of labeled formulas can be derivable from a set of sets of labeled formulas, using the tableau inference rules.

Definition 4.1.8 A set of labeled formulas Θ is *derivable* from a set $\mathcal{H}$ of sets of labeled formulas in S, denoted by

$$\mathcal{H} \vdash_S \Theta$$

if there is a finite sequence $\Psi_1 \dots \Psi_n$ of finite sets of labeled formulas such that:

- Ψ_1 is Θ;
- for each $i = 1,\dots,n$, one of the following holds:
  - there is a $\psi \in L(C)$ such that $1\!:\!\psi, 0\!:\!\psi \in \Psi_i$;
  - $\Psi_i \in \mathcal{H}$;
  - there exists a substitution σ and a tableau inference rule $\langle \Upsilon, \mu\rangle$ in S such that $\sigma(\mu) \in \Psi_i$ and $\sigma(\upsilon) \cup (\Psi_i \setminus \{\sigma(\mu)\}) \in \{\Psi_{i+1},\dots,\Psi_n\}$, for each $\upsilon \in \Upsilon$.

The sequence $\Psi_1 \dots \Psi_n$ is a *derivation* of Θ from $\mathcal{H}$ em S. ▿

We say that a formula φ is *derivable* from the set of formulas Γ in S, indicated by

$$\Gamma \vdash_S \varphi$$

if $\vdash_S \{(1:\delta) : \delta \in \Delta\} \cup \{0:\varphi\}$, where Δ is a finite subset of Γ. Again, $\vdash_S \varphi$ is used whenever Γ is an empty set. We denote by $\Gamma^{\vdash_S}$ the set of formulas derivable from Γ in S. When presenting a derivation, we add a justification for each set of formulas Ψ_i in the sequence: Abs (absurd) when $1:\psi, 0:\psi \in \Psi_i$, Hyp when it is a set in $\mathcal{H}$ and the name of the inference rule in the other cases.

Example 4.1.9 A tableau calculus $S_{\mathrm{P}_{\wedge,\Rightarrow}} = \langle C, R\rangle$ for the propositional connectives $\wedge$ and $\Rightarrow$ is such that

- $C_0 = \mathbb{P}$, $C_2 = \{\wedge, \Rightarrow\}$
- R, besides the excluded middle rule, includes

$$\frac{\{1:\xi_1, 1:\xi_2\}}{1:(\xi_1 \wedge \xi_2)}\ 1\wedge \qquad \frac{\{0:\xi_1\} \quad \{0:\xi_2\}}{0:(\xi_1 \wedge \xi_2)}\ 0\wedge$$

$$\frac{\{0:\xi_1\} \quad \{1:\xi_2\}}{1:(\xi_1 \Rightarrow \xi_2)}\ 1\Rightarrow \qquad \frac{\{1:\xi_1, 0:\xi_2\}}{0:(\xi_1 \Rightarrow \xi_2)}\ 0\Rightarrow$$

Observe that $0\wedge$ states that there are two alternatives for a conjunction to be false.

As an example, we present a derivation of $\{0:(((\xi_1 \Rightarrow \xi_2) \wedge \xi_1) \Rightarrow (\xi_1 \wedge \xi_2))\}$ from the empty set of hypothesis. Again, it is more convenient to read the derivation from the bottom to the top.

1	$\{0:(((\xi_1 \Rightarrow \xi_2) \wedge \xi_1) \Rightarrow (\xi_1 \wedge \xi_2))\}$	$0\Rightarrow$ 2
2	$\{1:((\xi_1 \Rightarrow \xi_2) \wedge \xi_1), 0:(\xi_1 \wedge \xi_2)\}$	$1\wedge$ 3
3	$\{1:(\xi_1 \Rightarrow \xi_2), 1:\xi_1, 0:(\xi_1 \wedge \xi_2)\}$	$0\wedge$ 4,5
4	$\{1:(\xi_1 \Rightarrow \xi_2), 1:\xi_1, 0:\xi_1\}$	Abs
5	$\{1:(\xi_1 \Rightarrow \xi_2), 1:\xi_1, 0:\xi_2\}$	$1\Rightarrow$ 6,7
6	$\{0:\xi_1, 1:\xi_1, 0:\xi_2\}$	Abs
7	$\{1:\xi_2, 1:\xi_1, 0:\xi_2\}$	Abs

Hence, we conclude that $\vdash_{S_{\mathrm{P}_{\wedge,\Rightarrow}}} \{0:(((\xi_1 \Rightarrow \xi_2) \wedge \xi_1) \Rightarrow (\xi_1 \wedge \xi_2))\}$ and therefore we also conclude that $\vdash_{S_{\mathrm{P}_{\wedge,\Rightarrow}}} (((\xi_1 \Rightarrow \xi_2) \wedge \xi_1) \Rightarrow (\xi_1 \wedge \xi_2))$.

Derivations in tableau calculi can be represented by trees. Again we choose to consider sequences for representing derivations since they are more convenient when dealing with fibring. The tree in Figure 4.1 corresponds to the derivation above. ∇

As in the case of Hilbert calculi and sequent calculi, each tableau calculus induces a consequence system.

Proposition 4.1.10 *A tableau calculus $S = \langle C, R\rangle$ induces a compact and structural consequence system $\mathcal{C}(S) = \langle C, \vdash_S\rangle$.*

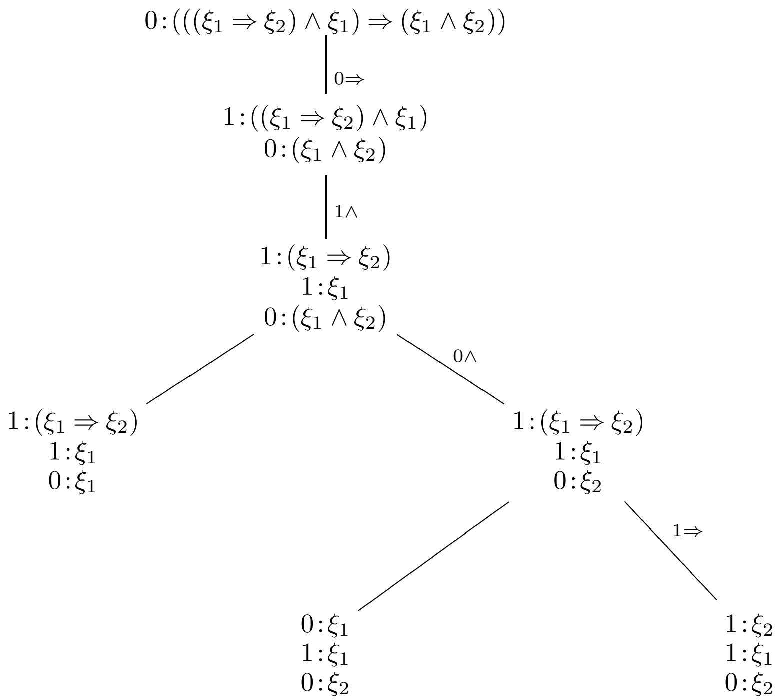


Figure 4.1: Tree for a derivation in $S_{\mathrm{P}_{\wedge,\Rightarrow}}$

Satisfaction systems

The semantic structures to be considered are satisfaction systems. These structures constitute a generalization of the interpretation systems defined in Section 3.1 of Chapter 3. Indeed they are more abstract and do not include details on the denotation of the connectives.

Definition 4.1.11 A *satisfaction system* is a triple $\langle C, M, \Vdash \rangle$ where C is a signature, M is a class and $\Vdash \subseteq M \times L(C)$. ▽

The elements of the class M are called models and $\Vdash$ is the satisfaction relation. In the sequel, we write $\text{Mod}(\varphi)$ to denote the set $\{m \in M : m \Vdash \varphi\}$ and $\text{Mod}(\Gamma)$ to denote the set $\bigcap_{\gamma \in \Gamma} \text{Mod}(\gamma)$, for $\Gamma \cup \{\varphi\} \subseteq L(C)$.

Example 4.1.12 A (Kripke) satisfaction system $Sat_{\mathbf{B}} = \langle C, M, \Vdash \rangle$ for modal logic **B** is as follows:

- C is the modal signature presented in Example 2.1.4;
- each model is a tuple (Kripke structure) $\langle W, R, V \rangle$ where W is a non-empty set, $R \subseteq W \times W$ is a symmetric and transitive relation and $V : \Xi \to \wp W$ is a map;
- $m \Vdash \varphi$ if $m, w \Vdash \varphi$ for every $w \in W$, where:
  - $m, w \Vdash \xi$ if $w \in V(\xi)$;
  - $m, w \Vdash (\neg\, \varphi)$ if not $m, w \Vdash \varphi$;
  - $m, w \Vdash (\varphi_1 \Rightarrow \varphi_2)$ if not $m, w \Vdash \varphi_1$ or $m, w \Vdash \varphi_2$;
  - $m, w \Vdash (\Box\varphi)$ if $m, u \Vdash \varphi$ for every $u \in W$ such that wRu. ▽

Observe that an interpretation system generates a satisfaction system. Moreover if we consider local and global reasoning then we can say that an interpretation system generates two satisfaction systems: a local and a global related to each other.

Example 4.1.13 Recall from Definition 3.1.16 that an interpretation system is a pair $\langle C, \mathcal{A} \rangle$ where C is a signature and $\mathcal{A}$ is a class of (algebraic) interpretation structures over C, that is, a class of tuples $\mathcal{B} = \langle B, \leq, \nu, \top \rangle$ where $\langle B, \leq, \top \rangle$ is a partial order with top $\top$ and $\langle B, \nu \rangle$ is an algebra over C.

Each interpretation system $I = \langle C, \mathcal{A} \rangle$ induces the satisfaction system

$$Sat_I = \langle C, M, \Vdash \rangle$$

as follows:

- each model in M is a pair $\langle \mathcal{B}, \alpha \rangle$ where $\mathcal{B}$ is an interpretation structure in $\mathcal{A}$ and α is an assignment of $\mathcal{B}$;

- $\langle \mathcal{B}, \alpha \rangle \Vdash \varphi$ if $[\![\varphi]\!]_{\mathcal{B}}^{\alpha} = \top$. ∇

Each satisfaction system induces a consequence system.

Proposition 4.1.14 *The satisfaction system* $\mathrm{Sat} = \langle C, M, \Vdash \rangle$ *induces a consequence system* $\mathcal{C}(\mathrm{Sat}) = \langle C, \vDash \rangle$ *where* $\Gamma^{\vDash} = \{\varphi \in L(C) : \mathit{Mod}(\Gamma) \subseteq \mathit{Mod}(\varphi)\}$ *for each* $\Gamma \subseteq L(C)$.

Proof. Let $\Gamma, \Gamma_1, \Gamma_2 \subseteq L(C)$.

Extensiveness: If $\varphi \in \Gamma$, then $\mathrm{Mod}(\Gamma) \subseteq \mathrm{Mod}(\varphi)$ and therefore $\varphi \in \Gamma^{\vDash}$.

Monotonicity: If $\Gamma_1 \subseteq \Gamma_2$ and $\varphi \in \Gamma_1^{\vDash}$ then $\mathrm{Mod}(\Gamma_2) \subseteq \mathrm{Mod}(\Gamma_1)$ and $\mathrm{Mod}(\Gamma_1) \subseteq \mathrm{Mod}(\varphi)$, therefore $\mathrm{Mod}(\Gamma_2) \subseteq \mathrm{Mod}(\varphi)$, hence $\varphi \in \Gamma_2^{\vDash}$.

Idempotence:

(i) We have $\mathrm{Mod}(\Gamma) \subseteq \mathrm{Mod}((\Gamma^{\vDash})^{\vDash})$. In fact, if $m \in \mathrm{Mod}(\Gamma)$ then $m \in \mathrm{Mod}(\gamma)$ for each $\gamma \in \Gamma^{\vDash}$, that is, $m \in \mathrm{Mod}(\Gamma^{\vDash})$. In particular, $\mathrm{Mod}(\Gamma^{\vDash}) \subseteq \mathrm{Mod}((\Gamma^{\vDash})^{\vDash})$. From this, the result follows easily.

(ii) Given $\varphi \in (\Gamma^{\vDash})^{\vDash}$, we have that $\mathrm{Mod}(\Gamma) \subseteq \mathrm{Mod}((\Gamma^{\vDash})^{\vDash}) \subseteq \mathrm{Mod}(\varphi)$, that is $\varphi \in \Gamma^{\vDash}$. $\triangleleft$

In order to guarantee that a satisfaction system induces a structural consequence system, an additional requirement is in order.

We say that a satisfaction system is *sensible-to-substitution* if for each substitution σ there is a map $\beta_\sigma : M \to M$ such that $m \Vdash \sigma(\varphi)$ if $\beta_\sigma(m) \Vdash \varphi$.

Example 4.1.15 Consider again the satisfaction system for modal logic as introduced in Example 4.1.12. We show that it is sensible-to-substitution. Let $\langle W, R, V \rangle$ be a model and $\sigma : \Xi \to L(C)$ a substitution. Then m_σ is the model $\langle W, R, V_\sigma \rangle$ where $V_\sigma(\xi) = \{w \in W : m, w \Vdash \sigma(\xi)\}$. Then, we can show by induction on φ that $m \Vdash \sigma(\varphi)$ if and only if $m_\sigma \Vdash \varphi$. ∇

Proposition 4.1.16 *Let* $\mathrm{Sat} = \langle C, M, \Vdash \rangle$ *be a sensible-to-substitution satisfaction system. Then* $\mathcal{C}(\mathrm{Sat})$ *is a structural consequence system.*

Proof. Given a substitution σ and a set of formulas Γ, we want to prove that $\sigma(\Gamma^{\vDash}) \subseteq (\sigma(\Gamma))^{\vDash}$. This is equivalent to prove that $\mathrm{Mod}(\sigma(\Gamma)) \subseteq \mathrm{Mod}(\varphi)$, for every $\varphi \in \sigma(\Gamma^{\vDash})$. Thus, let $\varphi \in \sigma(\Gamma^{\vDash})$. Given $m \in \mathrm{Mod}(\sigma(\Gamma))$, we have that $\beta_\sigma(m) \in \mathrm{Mod}(\Gamma)$ since, for each $\gamma \in \Gamma$, $m \Vdash \sigma(\gamma)$ and therefore $\beta_\sigma(m) \Vdash \gamma$. Recalling that $\varphi \in \sigma(\Gamma^{\vDash})$, let $\psi \in \Gamma^{\vDash}$ such that $\varphi = \sigma(\psi)$. Then, $\beta_\sigma(m) \Vdash \psi$ and therefore $m \Vdash \sigma(\psi)$, that is, $m \in \mathrm{Mod}(\varphi)$. Hence, $\mathrm{Mod}(\sigma(\Gamma)) \subseteq \mathrm{Mod}(\varphi)$ and, as a consequence, $\varphi \in (\sigma(\Gamma))^{\vDash}$. We then conclude that $\sigma(\Gamma^{\vDash}) \subseteq (\sigma(\Gamma))^{\vDash}$. $\triangleleft$

Usually it is not possible to prove that a consequence operator induced by a satisfaction system is compact. When dealing with logics that are complete, the compactness of the semantic entailment comes from the compactness of the syntactic consequence operator.

4.1.2 Fibring of consequence systems

We now define fibring of consequence systems. The signature of the fibring of two consequence systems is the fibring $C' \cup C''$ of the signatures C' and C'' of the two systems. Recall from Definition 2.1.7 that this corresponds to the union of the connectives in C' and C''. The language of the fibring of these consequence systems is the language over such signature. However, an essential ingredient for the definition of the consequence relation will be the ability to translate formulas of the fibring to either component. We achieve this by renaming the schema variables and coding formulas by fresh schema variables.

Definition 4.1.17 Let C and C' be two signatures such that $C \leq C'$ and consider a bijection $g : L(C') \to \mathbb{N}$. The translation

$$\tau_g : L(C') \to L(C)$$

is the map defined inductively as follows:

- $\tau_g(\xi_i) = \xi_{2i+1}$ for $\xi_i \in \Xi$;
- $\tau_g(c) = c$ for $c \in C_0$;
- $\tau_g(c') = \xi_{2g(c')}$ for $c' \in C'_0 \setminus C_0$;
- $\tau_g(c(\gamma'_1, \ldots, \gamma'_k)) = (c(\tau_g(\gamma'_1), \ldots, \tau_g(\gamma'_k)))$ for $c \in C_k$, $k > 0$ and $\gamma'_1, \ldots, \gamma'_k \in L(C')$;
- $\tau_g(c'(\gamma'_1, \ldots, \gamma'_k)) = \xi_{2g(c'(\gamma'_1,\ldots,\gamma'_k))}$ for $c' \in C'_k \setminus C_k$, $k > 0$ and formulas $\gamma'_1, \ldots, \gamma'_k \in L(C')$.

The substitution

$$\tau_g^{-1} : \Xi \to L(C')$$

is defined as

- $\tau_g^{-1}(\xi_{2i+1}) = \xi_i$ for $\xi_i \in \Xi$;
- $\tau_g^{-1}(\xi_{2i}) = g^{-1}(i)$. ∇

Observe that looking at the index of a variable in $\tau_g(L(C'))$ we can decide whether it comes from a variable or a formula starting with a connective in $C' \setminus C$.

As usual, we also denote by τ_g^{-1} the extension of τ_g^{-1} to $L(C)$.

Lemma 4.1.18 *Let C and C' be two signatures such that $C \leq C'$ and consider a bijection $g : L(C') \to \mathbb{N}$. The maps $\tau_g^{-1} \circ \tau_g$ and $\tau_g \circ \tau_g^{-1}$ are identity maps.*

Proof. Straightforward using induction. $\triangleleft$

Herein we choose to use $\vdash$ as a generic consequence operator instead of C for readability reasons.

In order to define the fibring of consequence systems $\langle C', \vdash' \rangle$ and $\langle C'', \vdash'' \rangle$, the relevant translations are defined from $L(C' \cup C'')$, the language of the fibring, to $L(C')$ and to $L(C'')$, the languages of each component. Given a bijection $g : L(C' \cup C'') \to \mathbb{N}$, the corresponding translation maps, defined using Definition 4.1.17, are

$$\tau'_g : L(C' \cup C'') \to L(C') \text{ and } \tau''_g : L(C' \cup C'') \to L(C'').$$

On the other hand, the corresponding substitutions are

$$\tau'^{-1}_g : \Xi \to L(C' \cup C'') \text{ and } \tau''^{-1}_g : \Xi \to L(C' \cup C'').$$

As usual the extension of τ'^{-1}_g to $L(C')$ is denoted by τ'^{-1}_g and the extension of τ''^{-1}_g to $L(C'')$ is denoted by τ''^{-1}_g.

In the sequel, for each pair of signatures C' and C'' we always assume fixed a bijection $g : L(C' \cup C'') \to \mathbb{N}$ and use τ', τ'', τ'^{-1} and τ''^{-1} instead of τ'_g, τ''_g, τ'^{-1}_g and τ''^{-1}_g.

The translations τ' and τ'', together with τ'^{-1} and τ''^{-1}, are used to define the closure of each $\Gamma \subseteq L(C' \cup C'')$ with respect to $\vdash'$ and to $\vdash''$.

Definition 4.1.19 Let $\mathcal{C}' = \langle C', \vdash' \rangle$ and $\mathcal{C}'' = \langle C'', \vdash'' \rangle$ be two consequence systems and let $\Gamma \subseteq L(C' \cup C'')$. The $\vdash'$-*closure* of Γ is the set

$$\Gamma^{\vdash'} = \tau'^{-1}((\tau'(\Gamma))^{\vdash'}).$$

Similarly, the $\vdash''$-*closure* of Γ is the set $\Gamma^{\vdash''} = \tau''^{-1}((\tau''(\Gamma))^{\vdash''})$. ▽

Next, we present the fibring of consequence systems.

Definition 4.1.20 Let $\mathcal{C}' = \langle C', \vdash' \rangle$ and $\mathcal{C}'' = \langle C'', \vdash'' \rangle$ be two consequence systems. The *fibring of consequence systems* $\mathcal{C}'$ and $\mathcal{C}''$ is a pair

$$\mathcal{C}' \cup \mathcal{C}'' = \langle C, \vdash \rangle$$

where

- $C = C' \cup C''$;
- $\vdash : \wp L(C) \to \wp L(C)$ where, for each $\Gamma \subseteq L(C)$, $\Gamma^{\vdash}$ is the least set satisfying the following:
  1. $\Gamma \subseteq \Gamma^{\vdash}$;
  2. If $\Delta \subseteq \Gamma^{\vdash}$ then $\Delta^{\vdash'} \cup \Delta^{\vdash''} \subseteq \Gamma^{\vdash}$. ▽

It is easy see that the set $\Gamma^{\vdash}$ can be constructed for every $\Gamma \subseteq L(C)$. In fact, given a set Γ of formulas, let

$$X = \{\Theta \in \wp L(C) : \Gamma \subseteq \Theta, \text{ and } \Delta \subseteq \Theta \text{ implies } \Delta^{\vdash'} \cup \Delta^{\vdash''} \subseteq \Theta\}.$$

Since $X \neq \emptyset$, because $L(C) \in X$, there exists the set $\bigcap X$. This set is $\Gamma^{\vdash}$.

As before, the fibring is said to be *unconstrained* when $C' \cap C'' = \emptyset$; otherwise is *constrained*.

Fibring of consequence systems can be seen as a "limit" construction over the class of quasi consequence systems. That is, consequences in general can be obtained by successive applications of one step consequences as we show in the next result using a Tarski's result on fixed points [257].

Proposition 4.1.21 *Let $\mathcal{C}' = \langle C', \vdash' \rangle$ and $\mathcal{C}'' = \langle C'', \vdash'' \rangle$ be consequence systems. Consider the following transfinite sequence of quasi consequence systems:*

- $\mathcal{C}_0 = \langle C' \cup C'', \vdash_0 \rangle$ *where* $\Gamma^{\vdash_0} = \Gamma$ *for every* $\Gamma \subseteq L(C)$*;*
- $\mathcal{C}_{\beta+1} = \langle C' \cup C'', \vdash_{\beta+1} \rangle$ *where* $\Gamma^{\vdash_{\beta+1}} = \tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \cup \tau''^{-1}((\tau''(\Gamma^{\vdash_\beta}))^{\vdash''})$ *for every* $\Gamma \subseteq L(C)$*;*
- $\mathcal{C}_\alpha = \langle C' \cup C'', \vdash_\alpha \rangle$ *where* $\Gamma^{\vdash_\alpha} = \bigcup_{\beta<\alpha} \Gamma^{\vdash_\beta}$ *if α is a limit ordinal.*

Then $\mathcal{C}' \cup \mathcal{C}'' = \mathcal{C}_\alpha$ for some ordinal α.

Proof. The operator $\Upsilon : \wp L(C) \to \wp L(C)$ such that $\Upsilon(\Delta) = \Delta^{\vdash'} \cup \Delta^{\vdash''}$ is monotonic and extensive (recall Definition 1.1.1) over the powerset complete lattice $\langle \wp L(C), \subseteq \rangle$. Hence Υ satisfies Tarski's fixed point theorem and so, for each Γ, there is a least fixed point $\Gamma^{\vdash_\alpha}$. It is easy to see that $\Gamma^{\vdash_\alpha} = \Gamma^{\vdash}$. ◁

Figure 4.2 illustrates a step of the fibring construction (just for the first component) where for simplicity:

- Ψ is $\Gamma^{\vdash_\beta}$;
- Ψ' is $\tau'(\Psi)^{\vdash'}$;
- Ψ^{+1} is $\Gamma^{\vdash_{\beta+1}}$.

Observe that the construction above can be transfinite. We can give a sufficient condition to ensure that the limit of the transfinite sequence defined above is obtained in a denumerable number of steps.

Proposition 4.1.22 *Let $\mathcal{C}' = \langle C', \vdash' \rangle$ and $\mathcal{C}'' = \langle C'', \vdash'' \rangle$ be compact consequence systems and consider the transfinite sequence of quasi consequence system as defined in Proposition 4.1.21. Then*

$$\mathcal{C}' \cup \mathcal{C}'' = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n.$$

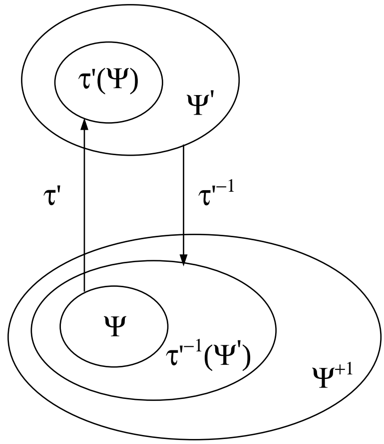


Figure 4.2: Construction of consequence system $\mathcal{C}_{\beta+1}$

Proof. It is enough to show that the operator $\Upsilon : \wp L(C) \to \wp L(C)$ defined in Proposition 4.1.21 is continuous with respect to the same order as before, and so Kleene's fixed point theorem can be applied. Recall that the operator Υ is continuous if it preserves directed unions. Let $\{\Delta_a\}_{a\in A}$ be a directed family of sets in $L(C)$. Monotonicity implies that $\bigcup_{a\in A}\Upsilon(\Delta_a) \subseteq \Upsilon(\bigcup_{a\in A}\Delta_a)$.

It remains to show the other inclusion. Let $\varphi \in \Upsilon(\bigcup_{a\in A}\Delta_a)$. Assume, without loss of generality, that $\varphi \in (\bigcup_{a\in A}\Delta_a)^{\vdash'}$. Then

$$\varphi \in \tau'^{-1}((\tau'(\bigcup_{a\in A}\Delta_a))^{\vdash'})$$

and so there is $\varphi' \in (\tau'(\bigcup_{a\in A}\Delta_a))^{\vdash'}$ such that $\tau'^{-1}(\varphi')$ is φ. Since $\mathcal{C}'$ is compact there is a finite $\Phi \subseteq (\tau'(\bigcup_{a\in A}\Delta_a))^{\vdash'}$ such that $\varphi' \in (\tau'(\Phi))^{\vdash'}$, hence, using monotonicity, there is $B \subseteq A$ finite such that

$$\varphi' \in (\tau'(\bigcup_{b\in B}\Delta_b))^{\vdash'}.$$

Since $\{\Delta_a\}_{a\in A}$ is directed there is $d \in A$ such that $\bigcup_{b\in B}\Delta_b = \Delta_d$. Therefore $\varphi \in \tau'^{-1}(\tau'(\Delta_d))$. Applying Kleene's fixed point theorem, $\Gamma^{\vdash} = \bigcup_{n\in\mathbb{N}}\Upsilon^n(\Gamma)$. $\triangleleft$

In general, we can still place an upper bound on the cardinality of α.

Proposition 4.1.23 *Consider* $\Gamma^{\vdash_0}, \Gamma^{\vdash_1}, \ldots, \Gamma^{\vdash_\alpha}$ *as defined in Proposition 4.1.21. Then* α *is countable.*

Proof. The sequence $\Gamma^{\vdash_0}, \Gamma^{\vdash_1}, \ldots, \Gamma^{\vdash_\alpha}$ is strictly increasing, hence $|\Gamma^{\vdash_\beta}| \geq |\beta|$ for each $\beta = 0, \ldots, \alpha$. Since $\Gamma^{\vdash} \subseteq L(C)$ and $L(C)$ is countable, it follows that α must also be countable. $\triangleleft$

In the sequel, we assume consequence systems $\mathcal{C}' = \langle C', \vdash' \rangle$ and $\mathcal{C}'' = \langle C'', \vdash'' \rangle$ and their fibring $\mathcal{C} = \langle C, \vdash \rangle$. We now show that the fibring of two consequence systems is indeed a consequence system.

Proposition 4.1.24 *The fibring $\mathcal{C}' \cup \mathcal{C}''$ of consequence systems $\mathcal{C}'$ and $\mathcal{C}''$ is a consequence system.*

Proof. Using Proposition 4.1.21, $\mathcal{C}' \cup \mathcal{C}''$ is a quasi-consequence system. Thus it is enough to prove idempotence.

Let $\Gamma \subseteq L(C)$. By definition of $\vdash$ there is α such that $(\Gamma^{\vdash})^{\vdash} = (\Gamma^{\vdash})^{\vdash_\alpha}$. We show by induction that

$$(\Gamma^{\vdash})^{\vdash_\alpha} \subseteq \Gamma^{\vdash} \text{ for every } \alpha.$$

(i) $\alpha = 0$. Straightforward.

(ii) $\alpha = \beta + 1$. By induction hypothesis $(\Gamma^{\vdash})^{\vdash_\beta} \subseteq \Gamma^{\vdash}$ and so, by definition of $\Gamma^{\vdash}$, we have $((\Gamma^{\vdash})^{\vdash_\beta})^{\vdash'} \cup ((\Gamma^{\vdash})^{\vdash_\beta})^{\vdash''} \subseteq \Gamma^{\vdash}$ which leads, by definition of $\vdash$, to $(\Gamma^{\vdash})^{\vdash_\alpha} \subseteq \Gamma^{\vdash}$.

(iii) α is a limit ordinal. Straightforward. $\triangleleft$

The following result shows that the closure in $\mathcal{C}' \cup \mathcal{C}''$ of a set of formulas Γ' in $L(C')$ is the same as the closure in $\mathcal{C}'$ of Γ', and the same applies to $\mathcal{C}''$, when we consider structural non-trivial consequence systems. As pointed out in [108], this is a key requirement for a good definition of fibring.

In the sequel, the following notion will be helpful.

Definition 4.1.25 A consequence system $\mathcal{C} = \langle C, \vdash \rangle$ is said to be *trivial (with respect to derivations)* if $\Gamma \vdash \varphi$ for every $\varphi \in L(C)$ and every non-empty $\Gamma \subseteq L(C)$. ∇

It is worth noting that, if $\mathcal{C}$ is structural, then $\mathcal{C}$ is trivial (with respect to derivations) if and only if there exist variables $\xi \neq \xi'$ such that $\xi' \vdash \xi$, if and only if there exist $\Xi' \cup \{\xi\} \subseteq \Xi$ such that $\Xi' \vdash \xi$ but $\xi \notin \Xi'$. Therefore, a structural consequence system $\mathcal{C}$ is non-trivial (with respect to derivations) if and only if, for every $\Xi' \cup \{\xi\} \subseteq \Xi$,

$$\Xi' \vdash \xi \;\text{ implies that }\; \xi \in \Xi'.$$

Proposition 4.1.26 *Unconstrained fibring $\mathcal{C}' \cup \mathcal{C}''$ of structural non-trivial (with respect to derivations) consequence systems is conservative, that is*

$$\Gamma'^{\vdash} \cap L(C', \Xi) = \Gamma'^{\vdash'}$$

for every $\Gamma' \subseteq L(C')$ *and*

$$\Gamma''^{\vdash} \cap L(C'', \Xi) = \Gamma''^{\vdash''}$$

for every $\Gamma'' \subseteq L(C'')$.

Proof. There is α such that $\Gamma'^{\vdash} = \Gamma'^{\vdash_\alpha}$. We show by induction that, for every α, $\Gamma'^{\vdash_\alpha} \cap L(C', \Xi) \subseteq \Gamma'^{\vdash'}$.

(i) $\alpha = 0$. Then $\Gamma' \subseteq \Gamma'^{\vdash'}$ by extensivity of $\vdash'$.

(ii) $\alpha = \beta + 1$. Let $\varphi' \in \Gamma'^{\vdash_\alpha}$. We have two cases.

(ii.a) $\varphi' \in \tau'^{-1}((\tau'(\Gamma'^{\vdash_\beta}))^{\vdash'}) \cap L(C', \Xi)$. Since $\mathcal{C}'$ is structural, it follows that

$$\tau'^{-1}((\tau'(\Gamma'^{\vdash_\beta}))^{\vdash'}) \subseteq (\tau'^{-1}(\tau'(\Gamma'^{\vdash_\beta})))^{\vdash'}$$

hence $\varphi' \in (\tau'^{-1}((\tau'(\Gamma'^{\vdash_\beta}))))^{\vdash'}$ and by Lemma 4.1.18, $\varphi' \in (\Gamma'^{\vdash_\beta})^{\vdash'}$. On the other hand, by the induction hypothesis, $\Gamma'^{\vdash_\beta} \subseteq \Gamma'^{\vdash'}$ and so, by monotonicity and idempotence of $\vdash'$, $\varphi \in \Gamma'^{\vdash'}$.

(ii.b) $\varphi' \in \tau''^{-1}((\tau''(\Gamma'^{\vdash_\beta}))^{\vdash''}) \cap L(C', \Xi)$. Then there is $\varphi'' \in (\tau''(\Gamma'^{\vdash_\beta}))^{\vdash''}$ such that $\tau''^{-1}(\varphi'')$ is φ'. By the induction hypothesis,

$$(\Gamma'^{\vdash_\beta}) \subseteq \Gamma'^{\vdash'} \subseteq L(C').$$

Recall the definition of τ'' from Definition 4.1.17. Since $C' \cap C'' = \emptyset$ and $(\Gamma'^{\vdash_\beta}) \subseteq L(C')$, it is clear that $\tau''(\Gamma'^{\vdash_\beta}) \subseteq \Xi$. Using again the fact that $C' \cap C'' = \emptyset$, and recalling the definition of the substitution τ''^{-1} from Definition 4.1.17, we obtain that φ'' must be a variable, because

$$\tau''^{-1}(\varphi'') = \varphi' \in L(C').$$

Since $\mathcal{C}''$ is structural, non-trivial and $\tau''(\Gamma'^{\vdash_\beta}) \cup \{\varphi''\} \subseteq \Xi$ is such that

$$\tau''(\Gamma'^{\vdash_\beta}) \vdash'' \varphi''$$

it follows that $\varphi'' \in \tau''(\Gamma'^{\vdash_\beta})$, by the remark above. Hence $\varphi' \in (\Gamma'^{\vdash_\beta})$ and so $\varphi' \in \Gamma'^{\vdash'}$, because $(\Gamma'^{\vdash_\beta}) \subseteq \Gamma'^{\vdash'}$.

(iii) α is a limit ordinal. Straightforward.

The proof that $\Gamma''^{\vdash} = \Gamma''^{\vdash''}$ for every $\Gamma'' \subseteq L(C'')$ is entirely analogous. ◁

If the consequence systems $\mathcal{C}'$ and $\mathcal{C}''$ are closed for renaming substitutions then they are both weaker than their fibring $\mathcal{C}' \cup \mathcal{C}''$. Recall that $\langle C', \vdash' \rangle$ is weaker than $\langle C, \vdash \rangle$ if $C' \leq C$ and $\Gamma'^{\vdash'} \subseteq \Gamma'^{\vdash}$ for every subset Γ' of $L(C')$.

Proposition 4.1.27 *For every consequence systems* $\mathcal{C}'$ *and* $\mathcal{C}''$ *closed for renaming substitutions the following relationships hold:* $\mathcal{C}' \leq \mathcal{C}' \cup \mathcal{C}''$ *and* $\mathcal{C}'' \leq \mathcal{C}' \cup \mathcal{C}''$

Proof. Since $C' \leq C' \cup C''$, it remains to show that $\Gamma'^{\vdash'} \subseteq \Gamma'^{\vdash}$ for $\Gamma' \subseteq L(C')$, and similarly for $\mathcal{C}''$. Assume that $\varphi' \in \Gamma'^{\vdash'}$. Let $\rho : \Xi \to L(C')$ be a renaming substitution such that

$$\rho(\xi_i) = \xi_{2i+1}$$

for every $\xi_i \in \Xi$. Since $\mathcal{C}'$ is closed for renaming substitutions, $\rho(\varphi') \in (\rho(\Gamma'))^{\vdash'}$. Observing that ρ coincides with τ' for formulas in $L(C')$, we have

$$\tau'(\varphi') \in (\tau'(\Gamma'))^{\vdash'}$$

hence $\varphi' \in \tau'^{-1}((\tau'(\Gamma'))^{\vdash'})$ and so $\varphi' \in \Gamma'^{\vdash}$. ◁

Theorem 4.1.28 *The fibring $\mathcal{C}' \cup \mathcal{C}''$ of structural consequence systems $\mathcal{C}'$ and $\mathcal{C}''$ is also structural.*

Proof. Let $\sigma : \Xi \to L(C)$ be a substitution. Since there is α such that $\Gamma^{\vdash} = \Gamma^{\vdash_\alpha}$, it is enough to show by induction that

$$\sigma(\Gamma^{\vdash_\alpha}) \subseteq (\sigma(\Gamma))^{\vdash} \text{ for every ordinal } \alpha.$$

(i) $\alpha = 0$. Then $\sigma(\Gamma^{\vdash_0}) = \sigma(\Gamma) \subseteq (\sigma(\Gamma))^{\vdash}$ by extensivity of $\vdash$.
(ii) $\alpha = \beta + 1$. We have to prove that $\sigma(\Gamma^{\vdash_{\beta+1}}) \subseteq (\sigma(\Gamma))^{\vdash}$, that is,

$$\sigma((\Gamma^{\vdash_\beta})^{\vdash'} \cup (\Gamma^{\vdash_\beta})^{\vdash''}) \subseteq (\sigma(\Gamma))^{\vdash}.$$

Now we prove that

$$\sigma((\Gamma^{\vdash_\beta})^{\vdash'}) \subseteq (\sigma(\Gamma))^{\vdash}.$$

The proof of $\sigma((\Gamma^{\vdash_\beta})^{\vdash''}) \subseteq (\sigma(\Gamma))^{\vdash}$ is similar. The induction hypothesis states that $\sigma(\Gamma^{\vdash_\beta}) \subseteq (\sigma(\Gamma))^{\vdash}$. Hence, following the definition of $\vdash$,

$$(\sigma(\Gamma^{\vdash_\beta}))^{\vdash'} \cup (\sigma(\Gamma^{\vdash_\beta}))^{\vdash''} \subseteq (\sigma(\Gamma))^{\vdash}$$

where $(\sigma(\Gamma^{\vdash_\beta}))^{\vdash'} = \tau'^{-1}(\tau'(\sigma(\Gamma^{\vdash_\beta}))^{\vdash'})$. Since $\tau' \circ \sigma : \Xi \to L(C')$ is a substitution and $\mathcal{C}'$ is structural,

$$\tau'^{-1}(\tau'(\sigma((\Gamma^{\vdash_\beta})^{\vdash'}))) \subseteq \tau'^{-1}((\tau'(\sigma(\Gamma^{\vdash_\beta})))^{\vdash'}).$$

Therefore, using Lemma 4.1.18,

$$\sigma((\Gamma^{\vdash_\beta})^{\vdash'}) \subseteq \tau'^{-1}((\tau'(\sigma(\Gamma^{\vdash_\beta})))^{\vdash'}).$$

We then conclude that $\sigma((\Gamma^{\vdash_\beta})^{\vdash'}) \subseteq (\sigma(\Gamma^{\vdash_\beta}))^{\vdash'} \subseteq (\sigma(\Gamma))^{\vdash}$.
(iii) α is a limit ordinal. Straightforward. ◁

Fibring plays a special role in the class of structural consequence systems: fibring of consequence systems $\mathcal{C}'$ and $\mathcal{C}''$ is minimal in the class of consequence systems that are stronger than $\mathcal{C}'$ and $\mathcal{C}''$.

Proposition 4.1.29 *For every structural consequence systems $\mathcal{C}'$, $\mathcal{C}''$ and $\mathcal{C}'''$, if $\mathcal{C}' \leq \mathcal{C}'''$ and $\mathcal{C}'' \leq \mathcal{C}'''$ then $\mathcal{C}' \cup \mathcal{C}'' \leq \mathcal{C}'''$.*

Proof. Let $\mathcal{C}''' = \langle C''', \vdash''' \rangle$. Clearly, $C' \cup C'' \leq C'''$. Hence, we have to prove that $\Gamma^{\vdash} \subseteq \Gamma^{\vdash'''}$. Since there is α such that $\Gamma^{\vdash} = \Gamma^{\vdash_\alpha}$, it is sufficient to show by induction that $\Gamma^{\vdash_\alpha} \subseteq \Gamma^{\vdash'''}$ for every α.

(i) $\alpha = 0$. Then $\Gamma \subseteq \Gamma^{\vdash'''}$, by extensivity of $\vdash'''$.

(ii) $\alpha = \beta + 1$. We have to show that

$$\tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \subseteq \Gamma^{\vdash'''}$$

and similarly $\tau''^{-1}((\tau''(\Gamma^{\vdash_\beta}))^{\vdash''}) \subseteq \Gamma^{\vdash'''}$. By the induction hypothesis $\Gamma^{\vdash_\beta} \subseteq \Gamma^{\vdash'''}$, hence $\tau'(\Gamma^{\vdash_\beta}) \subseteq \tau'(\Gamma^{\vdash'''})$ and so, by monotonicity of $\vdash'$,

$$(\tau'(\Gamma^{\vdash_\beta}))^{\vdash'} \subseteq (\tau'(\Gamma^{\vdash'''}))^{\vdash'}.$$

But $\mathcal{C}' \leq \mathcal{C}'''$ by hypothesis, hence

$$(\tau'(\Gamma^{\vdash'''}))^{\vdash'} \subseteq (\tau'(\Gamma^{\vdash'''}))^{\vdash'''}$$

and therefore $(\tau'(\Gamma^{\vdash_\beta}))^{\vdash'} \subseteq (\tau'(\Gamma^{\vdash'''}))^{\vdash'''}$. Moreover,

$$\tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \subseteq \tau'^{-1}((\tau'(\Gamma^{\vdash'''}))^{\vdash'''})$$

then, since $\mathcal{C}'''$ is structural,

$$\tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \subseteq (\tau'^{-1}(\tau'(\Gamma^{\vdash'''})))^{\vdash'''}.$$

Therefore $\tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \subseteq (\Gamma^{\vdash'''})^{\vdash'''}$ and so

$$\tau'^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \subseteq \Gamma^{\vdash'''}$$

by idempotence of $\vdash'''$.

(iii) α is a limit ordinal. Straightforward. ◁

From the results above, it follows that $\mathcal{C}' \cup \mathcal{C}''$ is the supremum of $\mathcal{C}'$ and $\mathcal{C}''$.

With respect to preservation of properties of consequence systems by fibring, we can prove the following.

Theorem 4.1.30 *The fibring of compact consequence systems is also a compact consequence system.*

Proof. We have to show that $\Gamma^{\vdash} = \bigcup_{\Phi \in \wp_{\text{fin}} \Gamma} \Phi^{\vdash}$.

The inclusion from right to left follows directly by extensivity and monotonicity.

In order to prove the converse inclusion, let $\varphi \in \Gamma^{\vdash}$. We prove by induction on α that if $\varphi \in \Gamma^{\vdash_\alpha}$ then there is $\Phi \subseteq \Gamma$ finite such that $\varphi \in \Phi^{\vdash}$.

(i) $\alpha = 0$. Then $\varphi \in \Gamma^{\vdash_0}$, hence $\varphi \in \Gamma$ and so we can take $\Phi = \{\varphi\}$.

(ii) $\alpha = \beta + 1$. We have two cases. Without loss of generality, let $\varphi \in (\Gamma^{\vdash_\beta})^{\vdash'}$. Since $\vdash'$ is compact there is $\Psi \subseteq \Gamma^{\vdash_\beta}$ finite such that $\varphi \in \Psi^{\vdash'}$ and, moreover, by definition of $\vdash$ also $\varphi \in \Psi^{\vdash}$. But, by the induction hypothesis, for each $\psi \in \Psi$ there is $\Phi_\psi \subseteq \Gamma$ finite such that $\psi \in \Phi_\psi^{\vdash}$. Take

$$\Phi = \bigcup_{\psi \in \Psi} \Phi_\psi.$$

Then Φ is a finite set such that $\Phi \subseteq \Gamma$ and $\varphi \in \Phi^{\vdash}$.

(iii) The case where α is a limit ordinal is straightforward. ◁

Now we can give a first attempt to solve the problem of heterogeneous fibring at the deductive level. The basic idea is that once we are given two calculi, of the same kind or not, we extract the induced consequence systems thus getting a homogeneous scenario. Afterwards we obtain the consequence system that represents the fibring of the induced consequence systems.

Example 4.1.31 Let $G_{\mathbf{S4}}$ be the sequent calculus for modal logic **S4** as presented in Example 4.1.4, but considering $C_1' = \{\neg, \Box'\}$, $C_2' = \{\Rightarrow\}$.

Let $\mathcal{C}(G_{\mathbf{S4}}) = \mathcal{C}' = \langle C', \vdash' \rangle$ be the induced consequence system. Let $H_{\mathbf{B}}$ be the Hilbert calculus for modal logic **B** as presented in Example 2.2.5 of Chapter 2, but considering $C_1'' = \{\neg, \Box''\}$, $C_2'' = \{\Rightarrow\}$, and let $\mathcal{C}(H_{\mathbf{B}}) = \langle C'', \vdash'' \rangle$ be the induced consequence system.

The fibring of these consequence systems is the consequence system

$$\mathcal{C}(G_{\mathbf{S4}}) \cup \mathcal{C}(H_{\mathbf{B}}) = (C' \cup C'', \vdash).$$

Note that the notion of derivation (finite sequence of either formulas or sequents) does not play a role in the construction.

We can conclude that $\Box''((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1)))) \in \emptyset^{\vdash}$ since

- $((\Box'\xi_3) \Rightarrow (\neg(\Box'(\neg\xi_3)))) \in \emptyset^{\vdash'}$
- $\tau'^{-1}(((\Box'\xi_3) \Rightarrow (\neg(\Box'(\neg\xi_3))))) = ((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1)))) \in \tau'^{-1}(\emptyset^{\vdash'})$
- $((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1)))) \in \emptyset^{\vdash_1}$ since $\tau'^{-1}(\emptyset^{\vdash'}) \subseteq \emptyset^{\vdash_1}$
- $\tau''(((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1))))) = (\xi_{2g(\Box'\xi_1)} \Rightarrow (\neg\, \xi_{2g(\Box'(\neg\, \xi_1))})) \in \tau''(\emptyset^{\vdash_1})$
- $\Box''((\xi_{2g(\Box'\xi_1)} \Rightarrow (\neg\, \xi_{2g(\Box'(\neg\, \xi_1))}))) \in (\tau''(\emptyset^{\vdash_1}))^{\vdash''}$
- $\tau''^{-1}(\Box''((\xi_{2g(\Box'\xi_1)} \Rightarrow (\neg\, \xi_{2g(\Box'(\neg\, \xi_1))})))) = (\Box''((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1)))))$
- $(\Box''((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1))))) \in \tau''^{-1}((\tau''(\emptyset^{\vdash_1}))^{\vdash''})$
- $(\Box''((\Box'\xi_1) \Rightarrow (\neg(\Box'(\neg\xi_1))))) \in \emptyset^{\vdash}$ since $\tau''^{-1}((\tau''(\emptyset^{\vdash_1}))^{\vdash''}) \subseteq \emptyset^{\vdash_2} \subseteq \emptyset^{\vdash}$. ▽

From a deductive point of view this solution to heterogeneous fibring is not entirely acceptable because the central notion of derivation as a finite sequence is lost. We do not have the notion of derivation in the fibring. Therefore we prefer to introduce, in the next section, the new notion of proof system to solve the problem of fibring heterogeneous proof system.

However, fibring of consequence systems is a good abstraction if we want to combine a semantic system with a deductive calculus as in the following example.

Example 4.1.32 Consider again the sequent calculus $G_{\mathbf{S4}}$ and the corresponding induced consequence system as presented in Example 4.1.31. Let $Sat_{\mathbf{B}}$ be the satisfaction system similar to the one introduced in Example 4.1.12 but with $C_1'' = \{\neg, \Box''\}$ and $C_2'' = \{\Rightarrow\}$. Let $\mathcal{C}(Sat_{\mathbf{B}}) = \langle C'', \vDash\rangle$ be the induced consequence system. The fibring of $G_{\mathbf{S4}}$ and $Sat_{\mathbf{B}}$ is the consequence system

$$\mathcal{C}(G_{\mathbf{S4}}) \cup \mathcal{C}(Sat_{\mathbf{B}}) = \langle C' \cup C'', \vdash\rangle.$$

Using the usual abbreviation $\Diamond'$ for simplicity, we are able to show that $(\Box'(\Box''(\Box''(\Diamond''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))) \in \emptyset^{\vdash}$:

- $(\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3))) \in \emptyset^{\vdash'}$
- $\tau'^{-1}((\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))) = (\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))) \in \tau'^{-1}(\emptyset^{\vdash'})$
- $(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))) \in \emptyset^{\vdash_1}$ since $\tau'^{-1}(\emptyset^{\vdash'}) \subseteq \emptyset^{\vdash_1}$;
- $\tau''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))) = \xi_{2i} \in \tau''(\emptyset^{\vdash_1})$
- $(\Box''(\Box''(\Diamond''(\xi_{2i})))) \in \{\xi_{2i}\}^{\vDash}$ hence $(\Box''(\Box''(\Diamond''(\xi_{2i})))) \in (\tau''(\emptyset^{\vdash_1}))^{\vDash}$
- $\tau''^{-1}((\Box''(\Box''(\Diamond''(\xi_{2i}))))) = (\Box''(\Box''(\Diamond''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))$
- $(\Box''(\Box''(\Diamond''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))) \in \tau''^{-1}((\tau''(\emptyset^{\vdash_1}))^{\vDash})$
- $(\Box''(\Box''(\Diamond''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))) \in \emptyset^{\vdash_2}$ since $\tau''^{-1}((\tau''(\emptyset^{\vdash_1}))^{\vDash}) \subseteq \emptyset^{\vdash_2}$
- $\tau'((\Box''(\Box''(\Diamond''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))) = \xi_{2j} \in \tau'(\emptyset^{\vdash_2})$
- $(\Box'\xi_{2j}) \in \{\xi_{2j}\}^{\vdash'}$ hence $(\Box'\xi_{2j}) \in (\tau'(\emptyset^{\vdash_2}))^{\vdash'}$
- $\tau'^{-1}((\Box'\xi_{2j})) = (\Box'(\Box''(\Box''(\Diamond''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))) \in \tau'^{-1}((\tau'(\emptyset^{\vdash_2}))^{\vdash'})$
- $(\Box'(\Box''(\Box''(\Diamond''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))) \in \emptyset^{\vdash}$ since $\tau'^{-1}((\tau'(\emptyset^{\vdash_2}))^{\vdash'}) \subseteq \emptyset^{\vdash_3} \subseteq \emptyset^{\vdash}$

where $i = g((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))$ and $j = g((\Box''(\Box''(\Diamond''((\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))))$. ∇

It is worthwhile to observe that given two Hilbert calculi H' and H'' then

$$\mathcal{C}(H' \cup H'') = \mathcal{C}(H') \cup \mathcal{C}(H'').$$

That is, the consequence system $\mathcal{C}(H' \cup H'')$ generated by the fibring $H' \cup H''$ of the two Hilbert calculi is the same as the fibring $\mathcal{C}(H') \cup \mathcal{C}(H'')$ of the consequence systems $\mathcal{C}(H')$ and $\mathcal{C}(H'')$ generated by H' and H'', respectively. In other words, if we consider $\mathcal{C}$ to be a map from Hilbert calculi to consequence systems then we can say that $\mathcal{C}$ preserves fibring. The same preservation holds for sequent and tableau calculi.

Of course one may ask if fibring the consequence systems induced by two interpretation structures is the best solution for solving the problem of heterogeneous fibring at the semantic level. The answer is that there are better ways, namely following an algebraic semantic approach as presented before in Chapter 3.

We synthesize the properties of fibring of consequence systems as follows:

- *homogeneous combination mechanism*: both original logics are presented by consequence systems;

- *algorithmic combination of logics*: given the consequence systems for the original logics, we know how to define the consequence system that corresponds to their fibring, but in many cases the given logics have to be pre-processed (that is, the consequence systems have to be extracted);

- *canonical combination of logics*: the fibring is the minimal consequence system among those that are stronger then the original consequence systems.

4.2 Fibring abstract proof systems

This section is dedicated to abstract proof systems and their fibring. Abstract proof systems put in a general setting the usual syntactic presentations of logics keeping the notion of derivation or certificate. When the given logics correspond to calculi of different nature, proof systems provide a better setting for heterogeneous fibring than consequence systems.

We start by presenting the notion of abstract proof system. Then we show how Hilbert calculi, sequent calculi and tableau calculi induce proof systems. Next we define fibring of proof systems. Heterogeneous fibring of logics present by different kinds of calculi are then achieved in two steps: we first get the proof system induced by each calculi and then we consider their fibring.

Given a binary relation $R \subseteq A \times B$, we use the notation $R(a, b)$ to indicate that $(a, b) \in R$, or to say that $R(a, b) = 1$ when viewing the relation as a map $1_R : A \times B \to \{0, 1\}$.

4.2.1 Abstract proof systems

Herein we introduce the notion of proof system and refer some of its properties.

Definition 4.2.1 An *abstract proof system* is a tuple

$$\mathcal{P} = \langle C, D, \circ, P\rangle$$

where C is a signature, D is a set, $\circ : \wp(D)\times D \to D$ is a map and $P = \{P_\Gamma\}_{\Gamma\subseteq L(C)}$ is a family of relations $P_\Gamma \subseteq D \times L(C)$ satisfying the following properties (where $P_\Gamma(E,\Psi)$ holds if for every $\psi \in \Psi$ there is $e \in E$ such that $P_\Gamma(e,\psi)$ holds):

- right reflexivity: $P_\Gamma(D,\Gamma)$ for every $\Gamma \subseteq L(C)$;
- monotonicity: $P_{\Gamma_1} \subseteq P_{\Gamma_2}$ for every $\Gamma_1 \subseteq \Gamma_2 \subseteq L(C)$;
- compositionality: let $\Gamma \cup \{\varphi\} \subseteq L(C)$:
  - $\emptyset \circ d = d$ for every $d \in D$;
  - If $E \subseteq D$ is a non-empty set and there is $\Psi \in \wp L(C)$ such that $P_\Gamma(E,\Psi)$ and $P_\Psi(d,\varphi)$ hold then $P_\Gamma(E \circ d,\varphi)$ also holds;
- variable exchange: $P_\Gamma(D,\varphi) = P_{\rho(\Gamma)}(D,\rho(\varphi))$ for any renaming substitution ρ, that is, a substitution such that $\rho(\xi) \in \Xi$ for every $\xi \in \Xi$. ▽

In an abstract proof system, the set D can be seen as the set of possible *derivations*, $\circ$ is a constructor that returns a derivation given a set of derivations and a derivation and, $P_\Gamma(d,\psi)$ holds when d is a derivation of ψ from the set of formulas Γ. The clause right reflexivity imposes that there is a derivation from a set for each of its elements. The clause compositionality states that we can show that a formula is derived from a set of formulas by using lemmas, that is, the usual cut rule for sets can be used. The meaning of the other clauses is straightforward. Note that the last clause is weaker than requiring structurality.

In the sequel, we may write "proof system" instead of "abstract proof system".

A tuple $\mathcal{P} = \langle C, D, \circ, P\rangle$ is a *quasi-proof system* if all the properties of a proof system hold with the possible exception of compositionality.

A particular (though not so interesting) proof system is the one where $D = L(C)$ and $P_\emptyset(\varphi,\varphi)$ for every $\varphi \in L(C)$. Another example is when we consider $D = L(C)^*$ (that is, D is the set of all finite sequences of formulas), $P_\Gamma(w,\gamma)$ if $\gamma \in \Gamma$ and is the last element of w. We stress that D does not need to be related to C and to the formulas in $L(C)$; for instance, D can be the set of natural numbers. Other examples will be discussed below.

Proposition 4.2.2 *The following properties hold in a proof system:*

- *falsehood:*
  for every $\Gamma \cup \{\varphi\} \subseteq L(C)$,

$$P_\Gamma(\emptyset,\varphi) = 0;$$

- *monotonicity on the first argument:*
  for every $E_1 \subseteq E_2 \subseteq D$ *and* $\Gamma, \Psi \subseteq L(C)$,

$$P_\Gamma(E_1, \Psi) \leq P_\Gamma(E_2, \Psi),$$

- *anti-monotonicity on the second argument:*
  for every $E \subseteq D$, $\Gamma \subseteq L(C)$ *and* $\Psi_1 \subseteq \Psi_2 \subseteq L(C)$,

$$P_\Gamma(E, \Psi_2) \leq P_\Gamma(E, \Psi_1);$$

- *union:*
  for every $\Gamma, \Psi_1, \Psi_2 \subseteq L(C)$,

$$P_\Gamma(E, \Psi_1 \cup \Psi_2) = P_\Gamma(E, \Psi_1) \times P_\Gamma(E, \Psi_2).$$

Proof. All the properties follow directly from the definitions. ◁

Proof systems sometimes have more properties. A proof system is said to be *non-trivial* if $P_{\Xi'}(D, \xi) = 0$ for any $\Xi' \subseteq \Xi$ and $\xi \in \Xi \setminus \Xi'$. A proof system is *structural* if $P_\Gamma(D, \varphi) \leq P_{\sigma(\Gamma)}(D, \sigma(\varphi))$ for each substitution σ. A proof system is said to be *compact or finitary* if for every Γ and ψ there is $\Phi \subseteq \Gamma$ finite such that

$$P_\Gamma(D, \psi) \leq P_\Phi(D, \psi)$$

for every $\Gamma \subseteq L(C)$.

Proof systems can be pre-ordered. We say that $\mathcal{P} = \langle C, D, \circ, P\rangle$ is *weaker* than $\mathcal{P}' = \langle C', D', \circ', P'\rangle$, indicated by

$$\mathcal{P} \leq \mathcal{P}'$$

if $C \leq C'$ and $P_\Gamma(D, \varphi) \leq P'_\Gamma(D', \varphi)$ for every $\Gamma \cup \{\varphi\} \subseteq L(C)$. Weaker proof systems prove less formulas. Contrary to the former cases, the weakness relation between proof systems is just a pre-order, that is, anti-symmetry may not hold.

4.2.2 Induced proof systems

We now show how Hilbert calculi, sequent calculi and tableau induce proof systems. Observe that the semantic presentations of logics cannot in general be presented as proof systems because they lack the notion of derivation.

Hilbert calculi

Recall from Definition 2.2.2 that a Hilbert calculus, in its simpler version, is a pair $\langle C, R\rangle$ such that C is a signature and R is a set of Hilbert inference rules (pairs whose first component is a finite set of formulas and whose second component is a formula).

We need some auxiliary notation. Let $\pi(e)$ denote the last element of sequence $e \in L(C)^*$, let $\pi(E)$ be the set $\{\pi(e) : e \in E\}$ where $E \subseteq L(C)^*$ and let $d_E^{\pi(E)}$ be the sequence obtained by replacing in $d \in L(C)^*$ the last element $\pi(e)$ of sequence e (whenever it occurs in d) by the sequence e for every $e \in E$. Strictly speaking, this is not well-defined, since there may be several e with the same last element, but this is not a problem as long as one assumes that e is chosen uniformly, that is choosing the first derivation in a lexicographical ordering of $L(C)^*$. As an illustration, assume that

$$d = \varphi_1 \dots \varphi_i \psi_k \varphi_{i+1} \dots \varphi_n$$

and let

$$E = \{\psi_1 \dots \psi_k, \psi'_1 \dots \psi'_r \psi_k\}.$$

Assume that

$$\psi_1 \dots \psi_k \leq \psi'_1 \dots \psi'_r \psi_k$$

in the lexicographical ordering mentioned above. Then $d_E^{\pi(E)}$ is the sequence

$$\varphi_1 \dots \varphi_i \psi_1 \dots \psi_k \varphi_{i+1} \dots \varphi_n.$$

Let us consider a more general example: suppose that

$$d = \varphi_1 \dots \varphi_{i_1-1} \psi_1 \varphi_{i_1+1} \dots \varphi_{i_2-1} \psi_2 \varphi_{i_2+1} \dots \varphi_{i_k-1} \psi_k \varphi_{i_k+1} \dots \varphi_n$$

and

$$E = \{e_1^1, \dots, e_{j_1}^1, e_1^2, \dots, e_{j_2}^2, \dots, e_1^k, \dots, e_{j_k}^k, e_1^{k+1}, \dots, e_{j_{k+1}}^{k+1}, \dots, e_1^r, \dots, e_{j_r}^r\}$$

such that $\pi(e_m^i) = \psi_i$ for $1 \leq m \leq j_i$ and $1 \leq i \leq r$, and ψ_i does not occur in d for $k < i \leq r$. Thus

$$\pi(E) = \{\psi_1, \psi_2, \dots, \psi_k, \psi_{k+1}, \dots, \psi_r\}.$$

Suppose also that $e_1^i \leq e_m^i$ for every $1 \leq m \leq j_i$ and $1 \leq i \leq k$ in the given lexicographical order, and let $e_1^i = \gamma_1^i \dots \gamma_{s_i}^i \psi_i$ for $1 \leq i \leq k$. Then

$$\begin{aligned} d_E^{\pi(E)} = {} & \varphi_1 \dots \varphi_{i_1-1} \gamma_1^1 \dots \gamma_{s_1}^1 \psi_1 \varphi_{i_1+1} \dots \varphi_{i_2-1} \gamma_1^2 \dots \gamma_{s_2}^2 \psi_2 \varphi_{i_2+1} \dots \\ & \dots \varphi_{i_k-1} \gamma_1^k \dots \gamma_{s_k}^k \psi_k \varphi_{i_k+1} \dots \varphi_n. \end{aligned}$$

Proposition 4.2.3 *A Hilbert calculus $H = \langle C, R\rangle$ induces a compact proof system $\mathcal{P}(H) = \langle C, D, \circ, P\rangle$ as follows:*

- $D = L(C)^*$;
- $E \circ d = d_E^{\pi(E)}$;
- $P_\Gamma(d, \psi)$ *holds if and only if d is a Hilbert-derivation for ψ from Γ.*

Proof. We have to check right reflexivity, monotonicity, compositionality and variable exchange.

Right reflexivity: It follows from the fact that the sequence γ is a derivation of γ from set Γ whenever $\gamma \in \Gamma$.

Monotonicity: Assume that d is a derivation of φ from Γ_1 and that $\Gamma_1 \subseteq \Gamma_2$. Then d is also a derivation of φ from Γ_2.

Compositionality: Assume that $P_\Gamma(E, \Psi)$ and $P_\Psi(d, \varphi)$ hold for some set Ψ. Then $d^\Psi_{E_\Psi}$, where $E_\Psi \subseteq E$ is the set of derivations in E of each $\psi \in \Psi$ from Γ, is a derivation of φ from Γ. Observe that only a finite number of elements of Ψ are used, so $d^\Psi_{E_\Psi}$ is still a finite sequence.

Variable exchange: Assume that d is a derivation of φ from Γ and that ρ is a renaming substitution. Then $\rho(d)$ is a derivation of $\rho(\varphi)$ from $\rho(\Gamma)$. ◁

It is easy to see that $\mathcal{P}(H)$ is also structural translating each derivation with the given substitution.

This is not the only proof system that can be generated from a Hilbert calculus. By construction any initial segment of a Hilbert derivation is itself a valid derivation, which motivates the following definition.

Example 4.2.4 Let $H = \langle C, R\rangle$ be a Hilbert calculus. Then $\mathcal{P}'(H)$ is defined as above, except that $P'_\Gamma(d, \varphi)$ now holds if and only if d is a valid derivation from Γ and φ occurs in d. ▽

The same example can be used to induce a proof system where the set of derivations bears no (apparent) relationship to the language.

Example 4.2.5 Let $H = \langle C, R\rangle$ be a Hilbert calculus. Let $g : L(C) \to \mathbb{N}$ be a Gödelization of $L(C)$ (that is, g is a bijection, there is an algorithm to evaluate g and an algorithm to evaluate g^{-1}) and $g^* : L(C)^* \to \mathbb{N}$ such that, for each $\varphi_1 \dots \varphi_k \in L(C)^*$,

$$g^*(\varphi_1 \dots \varphi_k) = p_1^{g(a_1)} \dots p_k^{g(a_k)}$$

where p_i is the ith prime number. Define $\mathcal{P}''(H)$ as follows:

- D is $\{g^*(e) : e \in L(C)^*\}$;
- $P_\Gamma(n, \varphi)$ if $n = g^*(e)$ of a sequence $e \in L(C)^*$ and e is a Hilbert-derivation of φ from Γ;
- $E \circ D = g^*(d_E^{\pi(E)})$. ▽

Sequent calculi

Recall that a sequent calculus is a pair $\langle C, R\rangle$ where C is a signature and R is a set of sequent inference rules (pairs whose first component is a finite set of sequents and whose second component is a sequent). In the sequel we assume that

R includes the Cut rule, the structural rules and, for each connective, a right and a left rule.

We need some auxiliary notation. Assume that d is a sequence of sequents with initial sequent $\Delta_1 \to \Delta_2$. When the initial sequent is important we can use $d_{\Delta_1 \to \Delta_2}$ to refer to d. Recall that in sequent calculi it is more convenient to read derivation from the bottom to the top.

Proposition 4.2.6 *A sequent calculus $G = \langle C, R \rangle$ induces a proof system $\mathcal{P}(G) = \langle C, D, \circ, P \rangle$ defined as follows:*

- $D = \mathrm{Seq}(C)^*$ *where* $\mathrm{Seq}(C)$ *is the set of all sequents defined with formulas in $L(C)$;*

- *Let $E \cup \{d_{\Theta \to \varphi}\} \subseteq D$ where $E \neq \emptyset$. Let $\{\theta_1, \dots, \theta_n\} \subseteq \Theta$ be the set of all formulas such that $d_{\Gamma_i \to \theta_i} \in E$ for every $i = 1, \dots, n$. Consider the set $\bar{\Theta} = \Theta \setminus \{\theta_1, \dots, \theta_n\}$. Then $E \circ d_{\Theta \to \varphi}$ is the following sequence:*

$$\begin{array}{lll}
 & \bar{\Theta}, \Gamma_1, \dots, \Gamma_n \to \varphi & \mathsf{Cut}\ 1a,1b \\
1a & \bar{\Theta}, \Gamma_1, \dots, \Gamma_n \to \varphi, \theta_1 & \mathsf{LW}^*, \mathsf{RW} \\
 & d_{\Gamma_1 \to \theta_1} & \\
1b & \theta_1, \bar{\Theta}, \Gamma_1, \dots, \Gamma_n \to \varphi & \mathsf{Cut}\ 2a,2b \\
 & \vdots & \\
 & \theta_1, \dots, \theta_{n-1}, \bar{\Theta}, \Gamma_1, \dots, \Gamma_n \to \varphi & \mathsf{Cut}\ na,nb \\
na & \theta_1, \dots, \theta_{n-1}, \bar{\Theta}, \Gamma_1, \dots, \Gamma_n \to \varphi, \theta_n & \mathsf{LW}^*, \mathsf{RW} \\
 & d_{\Gamma_n \to \theta_n} & \\
nb & \Theta, \Gamma_1, \dots, \Gamma_n \to \varphi & \mathsf{LW}^* \\
 & d_{\Theta \to \varphi} &
\end{array}$$

where $\mathsf{LW}^*, \mathsf{RW}$ *indicate several applications of left weakening followed by right weakening;*

- *$E \circ d_{\Delta_1 \to \Delta_2}$ is defined in a similar way;*

- *$\emptyset \circ d = d$;*

- *$P_\Gamma(d, \varphi)$ holds if d is a sequent-derivation of φ from Γ.*

Proof. We have to check right reflexivity, monotonicity, compositionality and variable exchange.

Right reflexivity: Just consider the derivation $\Gamma \to \gamma$ justified as an axiom whenever $\gamma \in \Gamma$.

Monotonicity: Consider a derivation $d_{\Gamma_1 \to \varphi}$ and $\Gamma_1 \subseteq \Gamma_2$. Then the following is a derivation for $\Gamma_2 \to \varphi$:

$$\begin{array}{lll}
1 & \Gamma_2 \to \varphi & \mathsf{LW}^*\ 2 \\
2 & d_{\Gamma_1 \to \varphi} &
\end{array}$$

Compositionality: Direct from the definition of $\circ$.

Variable exchange: If d is a derivation for $\Gamma \to \varphi$ and ρ is a renaming substitution then $\rho(d)$ is a derivation for $\rho(\Gamma) \to \rho(\varphi)$. $\triangleleft$

Note that, strictly speaking, the sequence $E \circ d_{\Theta \to \varphi}$ is not well-defined above, since there may be several derivations in E whose first sequent is the same. As remarked before in a similar situation, this is not a problem as long as one assumes an uniform choice of derivations.

Remark 4.2.7 Observe that in the case of sequents we can define binary relations $\overline{P}_{\mathcal{H}} \subseteq D \times \mathrm{Seq}(C)$ where $\mathcal{H}$ is a set of sequents over $L(C)$ (hence in $\mathrm{Seq}(C)$) but stating that $P_{\mathcal{H}}(d, s) = 1$ whenever $\mathcal{H} \vdash_G s$ with sequent-derivation d. Of course $P_\Gamma(d, \varphi)$ is $\overline{P}_\emptyset(d, \Gamma \to \varphi)$. $\triangledown$

Tableau calculi

Recall that a tableau calculus is a pair $\langle C, R\rangle$ where C is a signature and R is a set of tableau inference rules (pairs whose first component is a set of finite sets of labeled formulas and whose second element is a labeled formula). In the sequel we assume that R includes the excluded middle and for each connective a positive and a negative rule.

We need some notation: if $d_1 \ldots d_n$ is a finite sequence of sets and Ψ is a set of labeled formulas, then

$$\Psi d_1 \ldots d_n$$

is the finite sequence $d_1 \cup \Psi \ldots d_n \cup \Psi$. Observe that if $d_1 \ldots d_n$ is a tableau derivation for $\Gamma \vdash_S \varphi$ then $\Psi d_1 \ldots d_n$ is a tableau-derivation for $\Gamma \cup \Psi \vdash_S \varphi$. We use $1{:}A$ to refer to the set $\{(1{:}a) : a \in A\}$.

Proposition 4.2.8 *A tableau calculus $G = \langle C, R\rangle$ induces a proof system*

$$\mathcal{P}(G) = \langle C, D, \circ, P\rangle$$

defined as follows:

- $D = (\wp L^\lambda(C))^*$ *is the set of all finite sequences of sets of labeled formulas;*
- *Let $E \cup \{d\} \subseteq D$ be such that the first set in d is $1 : \Theta \cup \{0 : \varphi\}$ and, for $\theta_i \in \Theta$, with $i = 1, \ldots, n$, there are $e_i \in E$ whose first set is $1{:}\Gamma_i \cup \{0{:}\theta_i\}$. Let $\bar{\Theta} = \Theta \setminus \{\theta_1, \ldots, \theta_n\}$ and $\Gamma = \bigcup_{i=1}^n \Gamma_i$. Define*

  $$E \circ d$$

  as follows, where Ψ_1 is $1{:}\bar{\Theta} \cup (1{:}\Gamma \setminus 1{:}\Gamma_1) \cup \{0{:}\varphi\}$ and Ψ_n is $1{:}\bar{\Theta} \cup (1{:}\Gamma \setminus 1{:}\Gamma_n) \cup \{1{:}\theta_1, \ldots, 1{:}\theta_{n-1}, 0{:}\varphi\}$:

	$1{:}\bar{\Theta}, 1{:}\Gamma, 0{:}\varphi$	*EM 1a,1b*
1a	$0{:}\theta_1, 1{:}\bar{\Theta}, 1{:}\Gamma, 0{:}\varphi$	
	$\Psi_1 e_1$	
1b	$1{:}\theta_1, 1{:}\Gamma, 0{:}\varphi$	*EM 2a,2b*
	$\vdots$	
	$1{:}\theta_1, \ldots, 1{:}\theta_{n-1}, 1{:}\Gamma, 0{:}\varphi$	*EM na,nb*
na	$1{:}\theta_1, \ldots, 0{:}\theta_n, 1{:}\Gamma, 0{:}\varphi$	
	Ψ_n	
nb	$1{:}\Theta, 0{:}\varphi$	
	$1{:}\Gamma d$	

- *$P_\Gamma(d, \varphi)$ holds if and only if d is a tableau derivation for φ from Γ.*

Proof. We have to check right reflexivity, monotonicity, compositionality and variable exchange.

Right reflexivity: The set $1{:}\Gamma \cup \{0{:}\gamma\}$ is an absurd when $\gamma \in \Gamma$.

Monotonicity: Let $\Gamma_1 \subseteq \Gamma_2$ and d be a tableau-derivation of φ from Γ_1. Then $(\Gamma_2 \setminus \Gamma_1)d$ is a tableau-derivation of φ from Γ_2.

Compositionality: Follows from the fact that $E \circ d$ is a tableau-derivation of φ from $\bar{\Theta} \cap \Gamma$ whenever d is a tableau-derivation from Θ and there is a tableau-derivation in E of θ_i from Γ_i for every $i = 1, \ldots, n$.

Variable exchange: If d is a tableau-derivation of φ from Γ then $\rho(d)$ is a tableau-derivation of $\rho(\varphi)$ from $\rho(\Gamma)$ for every renaming substitution ρ. $\triangleleft$

Remark 4.2.9 Observe that in the case of tableau calculi we can define binary relations $\overline{P}_H \subseteq D \times \wp L^\lambda(C)$ where $\mathcal{H}$ is a set of sets of labeled formulas over $L(C)$ (hence in $\wp L^\lambda(C)$) but stating that $P_\mathcal{H}(d, s) = 1$ whenever $\mathcal{H} \vdash_S s$ with tableau-derivation d. Of course $P_\Gamma(d, \varphi)$ is $\overline{P}_\emptyset(d, 1{:}\Gamma \cup \{0{:}\varphi\})$. ∇

4.2.3 Fibring

We now define fibring of proof systems. As expect, the signature of the fibring is the fibring of the signatures of the components.

Definition 4.2.10 The *fibring* of two proof systems $\mathcal{P}' = \langle C', D', \circ', P' \rangle$ and $\mathcal{P}'' = \langle C'', D'', \circ'', P'' \rangle$ is the tuple

$$\mathcal{P}' \cup \mathcal{P}'' = \langle C, D, \circ, P \rangle$$

defined as follows:

- $C = C' \cup C''$;
- $D = \bigcup_{n \in \mathbb{N}} D_n$ where the sequence D_n is defined as follows:
  - $D_0 = D' \cup D''$;

- $D_{n+1} = \{\langle E, d\rangle : E \subseteq D_n,\ d \in D' \cup D''\}$;

- $E \circ d = \langle E, d\rangle$ if $E \neq \emptyset$ and d otherwise;
- $P_\Gamma(d', \varphi)$ holds if $P'_{\tau'(\Gamma)}(d', \tau'(\varphi))$ holds for $d' \in D'$;
- $P_\Gamma(d'', \varphi)$ holds if $P''_{\tau''(\Gamma)}(d'', \tau''(\varphi))$ holds for $d'' \in D''$;
- $P_\Gamma(\langle E, d\rangle, \varphi)$ holds if there is a set $\Psi \in L(C)$ for which both $P_\Psi(d, \varphi)$ and $P_\Gamma(E, \Psi)$ hold. ∇

Notice that the second- and third-to-last cases are not mutually exclusive since D' and D'' need not be disjoint.

Before showing that the fibring of proof systems is a proof system we give the intuition behind the construction of the set of derivations and some examples. Take as an example $\langle\{d'\}, d''\rangle$; this is a derivation provided that d' is a derivation in D' and d'' is a derivation in D''. Such a derivation is only relevant when we use the relation P. Saying that

$$P_\emptyset(\langle\{d'\}, d''\rangle, c''(c'(\xi_1)))$$

holds means that:

- d'' is a derivation in D'' of $c''(\xi_k)$ where $\xi_k = \tau''(c'(\xi_1))$, assuming that we take the singleton $\{\xi_k\}$ as the set of hypotheses, in other words provided that $P''_{\{\xi_k\}}(d'', c''(\xi_k))$ holds;
- d' is a derivation in D' of $c'(\xi_3)$ taking the empty set as the set of hypotheses, in other words provided that $P'_\emptyset(d', c'(\xi_3))$ holds.

We now give some examples of fibring involving logics presented with different calculi.

Example 4.2.11 Consider the proof system $\mathcal{P}(G_{\mathbf{S4}})$ induced by the sequent calculus for modal logic **S4** as presented in Example 4.1.31 and the proof system $\mathcal{P}(H_{\mathbf{B}})$ induced by the Hilbert calculus for modal logic with axiom B as in the same Example. Recall that they share the propositional connectives, but in the fibring we have two necessitations: $\Box'$ as in **S4** (and consequently a corresponding diamond $\Diamond'$) and $\Box''$ as in **B** (and consequently a corresponding diamond $\Diamond''$). We can prove in $\mathcal{P}(G_{\mathbf{S4}}) \cup \mathcal{P}(H_{\mathbf{B}})$ that

$$P_\emptyset(\langle\{\langle\{d'_1\}, d''\rangle\}, d'_2\rangle, (\Box'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))))$$

holds. Indeed

- $P'_{\{\xi_i\}}(d'_2, (\Box'\xi_i))$ holds in $\mathcal{P}(G_{\mathbf{S4}})$ with derivation d'_2 as follows:

1	$\to (\Box'\xi_i)$	R$\Box'$ 2
2	$\to \xi_i$	Hyp

where ξ_i is $\tau'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))$;

- and we have to show that $P_\emptyset(\langle\{d_1'\}, d''\rangle, \Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))$ holds;

But

$$P_\emptyset(\langle\{d_1'\}, d''\rangle, (\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))$$

holds since

- $P''_{\{\xi_j\}}(d'', (\Box''(\Diamond''(\Box''(\xi_j)))))$ holds in $\mathcal{P}(H_{\mathbf{B}})$ with derivation d'' as follows:

1	ξ_j	Hyp
2	$(\Box''\xi_j)$	Nec1
3	$((\Box''\xi_j) \Rightarrow (\Box''(\Diamond''(\Box''\xi_j))))$	B
4	$(\Box''(\Diamond''(\Box''\xi_j)))$	MP2,3

where ξ_j is $\tau''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))$;

- and $P_\emptyset(d_1', \Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$ holds in $\mathcal{P}(G_{\mathbf{S4}})$ with derivation d_1' as follows:

1	$\to (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	R$\Diamond'$ 2
2	$\to (\xi_3 \Rightarrow (\Box'\xi_3)), (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	R$\Rightarrow$ 3
3	$\xi_3 \to (\Box'\xi_3), (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	R$\Box'$ 4
4	$\to \xi_3, (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	R$\Diamond'$ 5
5	$\to \xi_3, (\xi_3 \Rightarrow (\Box'\xi_3)), (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	R$\Rightarrow$ 6
6	$\xi_3 \to \xi_3, (\Box'\xi_3), (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))$	Ax

Hence d_2', d'', d_1' provide a derivation for $(\Box'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))$ without any hypotheses. Observe that the number of pairings in the derivation indicates the way we have to use the component proof systems. In the example above we have three pairings and we had to use the component proof systems three times.

We can present the derivation in the fibring of the formula

$$(\Box'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))$$

from the empty set as follows:

$$\dfrac{\dfrac{\dfrac{d_1'}{\vdash_{S4} (\Diamond'(\xi_3 \Rightarrow (\Box'\xi_3)))}}{\vdash_{S5} (\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))}\,\tau'^{-1} \quad \dfrac{\dfrac{d''}{\{\xi_j\} \vdash_B (\Box''(\Diamond''(\Box''\xi_j)))}}{\{(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))\} \vdash_{S5} \psi}\,\tau''^{-1} \quad \dfrac{\dfrac{d_2'}{\{\xi_i\} \vdash_{S4} (\Box'\xi_i)}}{\{\psi\} \vdash_{S5} \varphi}\,\tau'^{-1}}{\vdash_{S5} (\Box'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))}\,Cut$$

where

- ψ is $(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))$;

- φ is $(\Box'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))))))$;
- ξ_j is $\tau''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1)))$;
- ξ_i is $\tau'(\Box''(\Diamond''(\Box''(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))))))$;
- $S4$ is the consequence system generated by the abstract deductive system $\mathcal{P}(S4)$ (see details in Subsection 4.2.4);
- B is the consequence system generated by the abstract deductive system $\mathcal{P}(B)$;
- $S5$ is the fibring of $S4$ and B.

We can describe the above presentation of the derivation in the following way: in order to show that φ is a theorem in the fibring $S5$, it is enough (cut rule) to derive in $S5$:

(i) φ from ψ;

(ii) ψ from $(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))$;

(iii) $(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))$ from the empty set.

We obtain (i) by deriving, in $S4$, $(\Box'\xi_i)$ from ξ_i (derivation d_2'). We get (ii) by deriving, in B, $(\Box''(\Diamond''(\Box''\xi_j)))$ from ξ_j (derivation d''). Finally, we obtain (iii) by deriving, in $S4$, $(\Diamond'(\xi_1 \Rightarrow (\Box'\xi_1))$ from the empty set (derivation d_1'). We use the fact that every derivation in a component is translated into a corresponding derivation in the fibring. ∇

Example 4.2.12 Consider the propositional part of the proof system $\mathcal{P}(G_{\mathbf{S4}})$ in Example 4.1.4 (with negation and implication as connectives) and the proof system $\mathcal{P}(S_{\mathrm{P}_{\wedge,\Rightarrow}})$ induced by the tableau calculus $S_{\mathrm{P}_{\wedge,\Rightarrow}}$ in Example 4.1.9. We prove in $\mathcal{P}(G_{\mathbf{S4}}) \cup \mathcal{P}(S_{\mathrm{P}_{\wedge,\Rightarrow}})$ that

$$P_{\emptyset}(\langle\{d_1', d_2'\}, d''\rangle, ((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_2))))$$

holds. Indeed, taking $\xi_i = \tau''(\neg(\xi_1 \Rightarrow \xi_2))$ and $\xi_j = \tau''(\neg\,\xi_2)$, we have:

- $P''_{\{(\xi_i \Rightarrow \xi_1),(\xi_i \Rightarrow \xi_j)\}}(d'', (\xi_i \Rightarrow (\xi_1 \wedge \xi_j)))$ holds in $S_{\mathrm{P}_{\wedge,\Rightarrow}}$ with derivation d'' as follows:

1.	$1{:}(\xi_i \Rightarrow \xi_1), 1{:}(\xi_i \Rightarrow \xi_j), 0{:}(\xi_i \Rightarrow (\xi_1 \wedge \xi_j))$	0⇒ 2
2.	$1{:}(\xi_i \Rightarrow \xi_1), 1{:}(\xi_i \Rightarrow \xi_j), 1{:}\xi_i, 0{:}(\xi_1 \wedge \xi_j)$	1⇒ 3,4
3.	$0{:}\xi_i, 1{:}(\xi_i \Rightarrow \xi_j), 1{:}\xi_i, 0{:}(\xi_1 \wedge \xi_j)$	Ax
4.	$1{:}\xi_1, 1{:}(\xi_i \Rightarrow \xi_j), 1{:}\xi_i, 0{:}(\xi_1 \wedge \xi_j)$	1⇒ 5,6
5.	$1{:}\xi_1, 0{:}\xi_i, 1{:}\xi_i, 0{:}(\xi_1 \wedge \xi_j)$	Ax
6.	$1{:}\xi_1, 1{:}\xi_j, 1{:}\xi_i, 0{:}(\xi_1 \wedge \xi_j)$	0∧ 7,8
7.	$1{:}\xi_1, 1{:}\xi_j, 1{:}\xi_i, 0{:}\xi_1$	Ax
8.	$1{:}\xi_1, 1{:}\xi_j, 1{:}\xi_i, 0{:}\xi_j$	Ax

- $P_\emptyset(d_1', ((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow \xi_1))$ holds in $G_{\mathbf{S4}}$ with derivation d_1' as follows:

$$\begin{array}{lll} 1. & \to ((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow \xi_1) & \text{R}\Rightarrow 2 \\ 2. & (\neg(\xi_1 \Rightarrow \xi_2)) \to \xi_1 & \text{L}\neg\ 3 \\ 3. & \to (\xi_1 \Rightarrow \xi_2), \xi_1 & \text{R}\Rightarrow 4 \\ 4. & \xi_1 \to \xi_2, \xi_1 & \text{Ax} \end{array}$$

- $P_\emptyset(d_2', (\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\neg\,\xi_2))$ holds in $G_{\mathbf{S4}}$ with derivation d_2' as follows:

$$\begin{array}{lll} 1. & \to (((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\neg\,\xi_2)) & \text{R}\Rightarrow 2 \\ 2. & (\neg(\xi_1 \Rightarrow \xi_2)) \to (\neg\,\xi_2) & \text{L}\neg\ 3 \\ 3. & \to (\xi_1 \Rightarrow \xi_2), (\neg\,\xi_2) & \text{R}\Rightarrow 4 \\ 4. & \xi_1 \to \xi_2, (\neg\,\xi_2) & \text{R}\neg\ 5 \\ 5. & \xi_1, \xi_2 \to \xi_2 & \text{Ax} \end{array}$$

Hence $d''d_1'd_2'$ constitutes a derivation of $(\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_2)))$ with no hypotheses. We have two pairings but in one of them we have to produce two derivations because the corresponding set has two elements. ∇

Observe that proofs in the fibring correspond to several application of a cut-like rule.

Example 4.2.13 Consider Example 4.2.12. The derivation of

$$((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_2))))$$

from $\emptyset$ can be seen as the following cut-like rule:

- Premises:
  - $\emptyset \vdash ((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow \xi_1));$
  - $\emptyset \vdash (\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\neg\,\xi_2));$
  - $\{(\xi_i \Rightarrow \xi_1), (\xi_i \Rightarrow \xi_j)\} \vdash (\xi_i \Rightarrow (\xi_1 \wedge \xi_j)));$
- Conclusion:
  - $\emptyset \vdash ((\neg(\xi_1 \Rightarrow \xi_2)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_2))))$

  where $\xi_i = \tau''(\neg(\xi_1 \Rightarrow \xi_2))$ and $\xi_j = \tau''(\neg\,\xi_2)$. ∇

We now prove that the fibring of proof systems is indeed a proof system.

Proposition 4.2.14 *The fibring $\mathcal{P}' \cup \mathcal{P}''$ of proof systems $\mathcal{P}'$ and $\mathcal{P}''$ is a proof system.*

Proof. We have to check right reflexivity, monotonicity, compositionality and variable exchange.

Right reflexivity: If $\varphi \in \Gamma$, then $\tau'(\varphi) \in \tau'(\Gamma)$, hence $P'_{\tau'(\Gamma)}(d', \tau'(\varphi))$ holds for some $d' \in D'$. Therefore $P_\Gamma(d', \varphi)$ also holds, so $P_\Gamma(D, \varphi)$ holds.

Monotonicity: Suppose $\Gamma_1 \subseteq \Gamma_2$ and suppose that $P_{\Gamma_1}(D, \varphi)$ holds. Then there is α such that $P_{\Gamma_1}(D_\alpha, \varphi)$ holds. We show by induction on α that $P_{\Gamma_2}(D_\alpha, \varphi)$ holds.

(i) $\alpha = 0$. Without loss of generality, assume that there is $d' \in D'$ such that $P'_{\tau'(\Gamma_1)}(d', \tau'(\varphi))$ holds. By monotonicity of $\mathcal{P}'$, also $P'_{\tau'(\Gamma_2)}(d', \tau'(\varphi))$ holds, thus $P_{\Gamma_2}(d', \varphi)$ holds and so $P_{\Gamma_2}(D_0, \varphi)$.

(ii) $\alpha = \beta + 1$. Assume $P_{\Gamma_1}(D_{\beta+1}, \varphi)$ holds. Hence there is Ψ such that $P_{\Gamma_1}(D_\beta, \Psi)$ and $P_\Psi(D, \varphi)$. Using the induction hypothesis we have $P_{\Gamma_2}(D_\beta, \Psi)$ and by definition of D we get $P_{\Gamma_2}(D_\alpha, \varphi)$.

(iii) α is a limit ordinal. This case is simple.

Compositionality: Immediate from the definition of $\circ$.

Variable exchange. Let ρ be a renaming substitution and suppose that $P_\Gamma(D, \varphi)$ holds. Then there is $n \in \mathbb{N}$ such that $P_\Gamma(D_n, \varphi)$ holds. We prove by induction on n that $P_{\rho(\Gamma)}(D_n, \rho(\varphi))$ holds.

Base: n=0. Suppose that d is $d' \in D'$; then

$$P'_{\tau'(\Gamma)}(d', \tau'(\varphi))$$

holds. We have to show that there is $e' \in D'$ such that

$$P'_{\tau'(\rho(\Gamma))}(e', \tau'(\rho(\varphi)))$$

holds. Consider the renaming substitution $\rho' : \Xi \to L(C')$ such that, for each ξ, $\rho'(\xi) = \tau'(\rho(\tau'^{-1}(\xi)))$. The variable exchange property for $\mathcal{P}'$ leads to the existence of $e' \in D$ such that

$$P'_{\rho'(\tau'(\Gamma))}(e', \rho'(\tau'(\varphi)))$$

holds. Since $\rho'(\tau'(\psi)) = \tau'(\rho(\psi))$ for every $\psi \in L(C)$ we conclude that

$$P'_{\tau'(\rho(\Gamma))}(e', \tau'(\rho(\varphi)))$$

holds. If d is $d'' \in D''$ the proof is similar.

Step: $n = k + 1$. Since $P_\Gamma(D_{k+1}, \varphi)$ then there is Ψ such that

$$P_\Gamma(D_k, \Psi) \text{ and } P_\Psi(D, \varphi).$$

For each $\psi \in \Psi$ there exists $e \in D_\beta$ for which $P_\Gamma(e, \psi)$ holds, and by induction hypothesis, there is some $e'(\psi) \in D_k$ (we use the notation $e'(\psi)$ to emphasize the existence of one for each ψ) for which $P_{\rho(\Gamma)}(e'(\psi), \rho(\psi))$ holds. Thus

$$P_{\rho(\Gamma)}(D_\beta, \rho(\Psi))$$

for $E' = \{e'(\psi) : \psi \in \Psi\}$. Using a reasoning similar to the one for the basis we conclude that $P_{\rho(\Psi)}(D, \rho(\varphi))$, and so $P_{\rho(\Gamma)}(D_\alpha, \rho(\varphi))$ holds. Straightforward. $\triangleleft$

Next we study the relationship between the fibring and the original proof systems showing that the latter are weaker than the former. Also of interest is to analyze how the fibring relates with proof systems that are stronger than the components.

Proposition 4.2.15 *For every proof systems $\mathcal{P}'$ and $\mathcal{P}''$ the following relationships hold: $\mathcal{P}' \leq \mathcal{P}' \cup \mathcal{P}''$ and $\mathcal{P}'' \leq \mathcal{P}' \cup \mathcal{P}''$.*

Proof. Since both situations are similar, we show the first one, which amounts to showing that $P'_{\Gamma'}(D', \varphi') \leq P_{\Gamma'}(D, \varphi')$ for $\Gamma' \cup \{\varphi'\} \subseteq L(C')$. Suppose that $P'_{\Gamma'}(d', \varphi')$ holds. Then, since τ' is a renaming substitution on $L(C')$, there is a derivation $d \in D'$ such that $P'_{\tau'(\Gamma)}(d, \tau'(\varphi))$ holds, and therefore $P_\Gamma(d, \varphi)$ holds. $\lhd$

We need an auxiliary result before characterizing fibring in the class of proof systems that are stronger than the components.

Theorem 4.2.16 *The fibring of structural proof systems is also structural.*

The proof of this result is similar to the one showing that the fibring satisfies variable exchange, and for this reason we omit it.

Proposition 4.2.17 *Let $\mathcal{P}'$, $\mathcal{P}''$ and $\mathcal{P}'''$ be structural proof systems. If $\mathcal{P}' \leq \mathcal{P}'''$ and $\mathcal{P}'' \leq \mathcal{P}'''$ then $\mathcal{P}' \cup \mathcal{P}'' \leq \mathcal{P}'''$.*

Proof. We have to show that $P_\Gamma(D, \varphi) \leq P'''_\Gamma(D''', \varphi)$. Assume that $P_\Gamma(D, \varphi)$ holds. Then there is $d \in D$ such that $P_\Gamma(d, \varphi)$. We prove by induction on d that there is d''' such that $P'''_\Gamma(d''', \varphi)$.

(i) Assume that d is $d' \in D'$. Then $P'_{\tau'(\Gamma)}(d', \tau'(\varphi))$ holds and so by the hypothesis on $\mathcal{P}'''$ there is $e''' \in D'''$ such that $P'''_{\tau'(\Gamma)}(e''', \tau'(\varphi))$ also holds. Since $\mathcal{P}'''$ is closed for substitution there is $d''' \in D'''$ such that $P'''_{\tau'^{-1}(\tau'(\Gamma))}(d''', \tau^{-1}(\tau'(\varphi)))$ holds and so there is $d''' \in D'''$ such that $P'''_\Gamma(d''', \varphi)$ holds. If d is $d'' \in D''$, the situation is analogous.

(ii) Assume that d is $\langle E, f\rangle$. Then there is $\Psi \subseteq L(C)$ such that $P_\Psi(f, \varphi)$ and $P_\Gamma(E, \Psi)$ hold. By induction hypothesis there is E''' such that $P'''_\Gamma(E''', \Psi)$ holds. Using a reasoning similar to the one above, there is $d''' \in D'''$ such that $P'''_\Psi(d''', \varphi)$ holds. Hence $P'''_\Gamma(E''' \circ d''', \varphi)$ holds. $\lhd$

Now we turn our attention towards preservation of compactness.

Theorem 4.2.18 *The fibring of compact proof systems is compact.*

Proof. We prove, by induction on d, that there is $\Phi \subseteq \Gamma$ finite such that $P_\Phi(d, \varphi)$ whenever $P_\Gamma(d, \varphi)$.

(i) Let $d \in D'$. Then $P'_{\tau'(\Gamma)}(d, \tau'(\varphi))$, so there are $\Phi' \subseteq \tau'(\Gamma)$ finite and $d' \in D'$ such that $P'_{\Phi'}(d', \tau'(\varphi))$ and so $P_{\tau'^{-1}(\Phi')}(d', \varphi)$ where $\tau'^{-1}(\Phi') \subseteq \Gamma$ is finite. The case $d \in D''$ is analogous.

(ii) Let $d = \langle E, d' \rangle$. Then there is Ψ such that $P_\Gamma(E, \Psi)$ and $P_\Psi(d', \varphi)$ hold. Using a reasoning similar to the one in the basis, we can conclude that there are $\Phi \subseteq \Psi$ finite and $f \in D$ such that $P_\Phi(f, \varphi)$ holds. On the other hand, since $P_\Gamma(E, \Phi)$ holds, then by induction hypothesis there are $\Gamma' \subseteq \Gamma$ finite and $F \subseteq D$ such that $P_{\Gamma'}(F, \Phi)$, and so $P_{\Gamma'}\langle F, f \rangle, \varphi)$. $\triangleleft$

As a consequence, the finite-derivation fibring of compact proof systems has the same deductive power as their fibring, that is, the value of $P_\Gamma(D, \varphi)$ is independent on whether D is obtained by fibring or by finite-derivation fibring.

We synthesize the properties of fibring of proof systems as follows:

- *homogeneous combination mechanism*: both original logics are presented by proof systems;
- *algorithmic combination of logics*: given the proof systems for the original logics, we know how to define the proof system that corresponds to their fibring, but in many cases the given logics have to be pre-processed (that is, the proof systems have to be extracted from the original logics);
- *canonical combination of logics*: the fibring is the minimal proof system among those that are stronger than the original proof systems.

4.2.4 Proof systems vs consequence systems

This subsection concentrates on the study of some relationship between proof systems and consequence systems. We start by discussing the generation of a consequence system out of a proof system.

Proposition 4.2.19 *A proof system $\mathcal{P} = \langle C, D, \circ, P \rangle$ induces a consequence system $\mathcal{C}(\mathcal{P}) = \langle C, \vdash \rangle$ where $\Gamma^\vdash = \{\varphi \in L(C) : P_\Gamma(D, \varphi)\}$.*

Proof. We have to check extensiveness, monotonicity and idempotence.

Extensiveness: Follows directly from the right reflexivity of P_Γ.

Monotonicity: Suppose that $\Gamma_1 \subseteq \Gamma_2$ and that $\varphi \in \Gamma_1^\vdash$. Then $P_{\Gamma_1}(D, \varphi)$ holds. By the monotonicity of $\mathcal{P}$ we have $P_{\Gamma_1}(D, \varphi) \leq P_{\Gamma_2}(D, \varphi)$ hence $P_{\Gamma_2}(D, \varphi)$ holds and so $\varphi \in \Gamma_2^\vdash$.

Idempotence: Suppose that $\varphi \in (\Gamma^\vdash)^\vdash$. Then there is $d \in D$ such that $P_{\Gamma^\vdash}(d, \varphi)$. On the other hand, there is $E \subseteq D$ such that $P_\Gamma(E, \Gamma^\vdash)$. Hence by compositionality in $\mathcal{P}$ we have $P_\Gamma(E \circ d, \varphi)$ and so $\varphi \in \Gamma^\vdash$. $\triangleleft$

We will now investigate how properties of the proof system are propagated to the induced consequence system.

Proposition 4.2.20 *If $\mathcal{P}$ is structural then $\mathcal{C}(\mathcal{P})$ is structural.*

The proof of the above result presents no major difficulties and is left to the reader.

In what concerns compactness we can obtain the following result.

Proposition 4.2.21 *If $\mathcal{P}$ is compact then $\mathcal{C}(\mathcal{P})$ is compact.*

Proof. Suppose that $\mathcal{P}$ is compact and $\varphi \in \Gamma^{\vdash}$. Then $P_\Gamma(D, \varphi)$ holds and so by, compactness of $\mathcal{P}$, there is $\Phi \subseteq \Gamma$ finite such that $P_\Phi(D, \varphi)$ also holds and so $\varphi \in \Phi^{\vdash}$. $\triangleleft$

Also of interest is the relationship between a calculus and the proof system it induces. We do a parametric proof of the following result.

Proposition 4.2.22 *Let Calc be a (Hilbert, sequent, tableau) calculus. Then $\mathcal{C}(\mathcal{P}(\mathit{Calc})) = \mathcal{C}(\mathit{Calc})$.*

Proof. Since both $\mathcal{C}(\mathcal{P}(\text{Calc}))$ and $\mathcal{C}(\text{Calc})$ share the same signature C, all one needs to show is that the closure of a set $\Gamma \subseteq L(C)$ is the same in both cases. Let $\mathcal{C}(\mathcal{P}(\text{Calc})) = \langle C, \vdash_1 \rangle$ and $\mathcal{C}(\text{Calc}) = \langle C, \vdash_2 \rangle$. Then $\varphi \in \Gamma^{\vdash_1}$ if and only if $P_\Gamma(D, \varphi)$ holds in $\mathcal{P}(\text{Calc})$ if and only if there is a Calc-derivation of φ from Γ in D if and only if $\varphi \in \Gamma^{\vdash_2}$. $\triangleleft$

The following result indicates that relationships between proof systems are preserved by the induced consequence systems.

Proposition 4.2.23 *Let P and P' be proof systems such that $P \leq P'$. Then $\mathcal{C}(P) \leq \mathcal{C}(P')$. The converse is also true.*

Proof. Let $\mathcal{P} \leq \mathcal{P}'$. Then $C \leq C'$. Let $\Gamma \subseteq L(C)$ and suppose that $\varphi \in \Gamma^{\vdash}$. Then $P_\Gamma(D, \varphi)$ holds, whence $P'_\Gamma(D', \varphi)$ also holds since $\mathcal{P} \leq \mathcal{P}'$ and so $\varphi \in \Gamma^{\vdash'}$. The proof of the converse is analogous. $\triangleleft$

As a special case we conclude $\mathcal{C}(\mathcal{P}') \leq \mathcal{C}(\mathcal{P}' \cup \mathcal{P}'')$ and $\mathcal{C}(\mathcal{P}'') \leq \mathcal{C}(\mathcal{P}' \cup \mathcal{P}'')$.

Now we show how to generate a proof system out of a consequence system closed for renaming substitutions.

Proposition 4.2.24 *A consequence system $\mathcal{C}$ closed for renaming substitutions induces a proof system $\mathcal{P}(\mathcal{C})$ with the same signature as follows:*

- $D = \{*\}$;
- $E \circ * = *$ *for* $E \subseteq D$;
- $P_\Gamma(*, \varphi)$ *holds if and only if* $\varphi \in \Gamma^{\vdash}$.

Proof. We have to check right reflexivity, monotonicity, compositionality and variable exchange.

Right reflexivity: Since $\vdash$ is extensive, $\Gamma \subseteq \Gamma^{\vdash}$ for every $\Gamma \subseteq L(C)$, so $P_\Gamma(D, \Gamma)$ holds.

Monotonicity: Assume that $\Gamma_1 \subseteq \Gamma_2$ and $P_{\Gamma_1}(D, \varphi)$ holds. Then $\varphi \in \Gamma_1^{\vdash}$; by monotonicity of $\mathcal{C}$, $\Gamma_1^{\vdash} \subseteq \Gamma_2^{\vdash}$, hence $\varphi \in \Gamma_2^{\vdash}$, and so $P_{\Gamma_2}(D, \varphi)$.

Compositionality: Suppose that $P_\Psi(d, \varphi)$ and, for non-empty $E \subseteq D$, $P_\Gamma(E, \Psi)$ hold. Then $\Psi \subseteq \Gamma^{\vdash}$ and $\varphi \in \Psi^{\vdash}$, hence, by monotonicity of $\mathcal{C}$, $\varphi \in (\Gamma^{\vdash})^{\vdash}$ and so, by idempotence of $\vdash$, $\varphi \in \Gamma^{\vdash}$. Therefore $P_\Gamma(E \circ d, \varphi)$ holds.

Variable exchange: Assume that ρ is a renaming substitution and that $P_\Gamma(D, \varphi)$ holds. Then $\varphi \in \Gamma^{\vdash}$, hence $\rho(\varphi) \in \rho(\Gamma)^{\vdash}$ and so $P_{\rho(\Gamma)}(D, \rho(\varphi))$. ◁

We can show easily that the induced proof system is structural and compact whenever the consequence system has the same properties. A proof system $\mathcal{P}$ can be compared with the proof system generated by the consequence system induced by $\mathcal{P}$ as the following result states.

Proposition 4.2.25 *For any proof system $\mathcal{P}$, $\mathcal{P} \leq \mathcal{P}(\mathcal{C}(\mathcal{P}))$ and $\mathcal{P}(\mathcal{C}(\mathcal{P})) \leq \mathcal{P}$.*

Proof. Straightforward. Since in both constructions the signature does not change, all that is left to show is that, $P_\Gamma(D, \varphi)$ holds in $\mathcal{P}$ if and only if $P_\Gamma(*, \varphi)$ holds in $\mathcal{P}(\mathcal{C}(\mathcal{P}))$. Thus, $P_\Gamma(D, \varphi)$ holds in $\mathcal{P}$ if and only if $\varphi \in \Gamma^{\vdash}$ if and only if $P_\Gamma(\{*\}, \varphi)$ holds in $\mathcal{P}(\mathcal{C}(\mathcal{P}))$. ◁

On the other hand, the opposite relation also holds.

Proposition 4.2.26 *For every consequence system $\mathcal{P}$, $\mathcal{C} = \mathcal{C}(\mathcal{P}(\mathcal{C}))$.*

Proof. Let $\mathcal{C}(\mathcal{P}(\mathcal{C})) = \langle C, \vdash' \rangle$ and let $\Gamma \cup \{\varphi\} \subseteq L(C)$. Then $\varphi \in \Gamma^{\vdash'}$ if and only if $P_\Gamma(D, \varphi)$ holds in $\mathcal{P}(\mathcal{C})$ if and only if $\varphi \in \Gamma^{\vdash}$. ◁

Finally, we relate the consequence system induced by the fibring of proof systems with the fibring of the consequence systems induced by the proof systems.

Proposition 4.2.27 *The fibring of proof systems has the following property:*

$$\mathcal{C}(\mathcal{P}' \cup \mathcal{P}'') = \mathcal{C}(\mathcal{P}') \cup \mathcal{C}(\mathcal{P}'').$$

Proof. The signature of both $\mathcal{C}(\mathcal{P}' \cup \mathcal{P}'')$ and $\mathcal{C}(\mathcal{P}') \cup \mathcal{C}(\mathcal{P}'')$ is $C = C' \cup C''$. Denoting $\mathcal{C}(\mathcal{P}' \cup \mathcal{P}'')$ by $\langle C, \vdash_a \rangle$ and $\mathcal{C}(\mathcal{P}') \cup \mathcal{C}(\mathcal{P}'')$ by $\langle C, \vdash_b \rangle$, all that is left to show is that

$$\Gamma^{\vdash_a} = \Gamma^{\vdash_b} \text{ for all } \Gamma \subseteq L(C).$$

(i) We start by showing that $\Gamma^{\vdash_a} \subseteq \Gamma^{\vdash_b}$. Suppose $\varphi \in \Gamma^{\vdash_a}$. Then $P_\Gamma(D, \varphi)$ holds, hence $P_\Gamma(d, \varphi)$ holds for some $d \in D$. We prove that $\varphi \in \Gamma^{\vdash_b}$ by induction on d.

(i.a) If d is $d' \in D'$, then

$$P'_{\tau'(\Gamma)}(d', \tau'(\varphi))$$

hence $\tau'(\varphi) \in \tau'(\Gamma)^{\vdash'}$ and therefore $\varphi \in \Gamma^{\vdash_b}$ by definition of fibring of consequence systems. The case where d is $d'' \in D''$ is analogous.

(i.b) If d is $\langle E, d''' \rangle$ with $E \cup \{d'''\} \subseteq D$, then there is a set Ψ such that

$$P_\Gamma(E, \Psi) \text{ and } P_\Psi(d''', \varphi)$$

both hold, that is, $\Psi \subseteq \Gamma^{\vdash_a}$ and $\varphi \in \Psi^{\vdash_a}$. By induction hypothesis, $\Psi \subseteq \Gamma^{\vdash_b}$ and $\varphi \in \Psi^{\vdash_b}$ and, by idempotence of $\vdash_b$, it follows that $\varphi \in (\Gamma^{\vdash_b})^{\vdash_b} \subseteq \Gamma^{\vdash_b}$.

(ii) Now we show that $\Gamma^{\vdash_b} \subseteq \Gamma^{\vdash_a}$. Suppose now that $\varphi \in \Gamma^{\vdash_b}$. Then $\varphi \in \Gamma^{\vdash_\beta}$ for some ordinal β in the fixed point construction of Proposition 4.1.21. We prove that $\varphi \in \Gamma^{\vdash_a}$ by induction on β.

(ii.a) $\beta = 0$. Straightforward, since $\Gamma \subseteq \Gamma^{\vdash_a}$.

(ii.b) If $\varphi \in \Gamma^{\vdash_{\beta+1}}$, then

$$\text{either } \varphi \in {\tau'}^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}) \text{ or } \varphi \in {\tau''}^{-1}((\tau''(\Gamma^{\vdash_\beta}))^{\vdash''}).$$

Both cases are similar, so assume the first one holds. By induction hypothesis $\Gamma^{\vdash_\beta} \subseteq \Gamma^{\vdash_a}$, so $P_\Gamma(D, \Gamma^{\vdash_\beta})$ holds. Also, from $\varphi \in {\tau'}^{-1}((\tau'(\Gamma^{\vdash_\beta}))^{\vdash'})$ we conclude that $\tau'(\varphi) \in (\tau'(\Gamma^{\vdash_\beta}))^{\vdash'}$, so

$$P'_{\tau'(\Gamma^{\vdash_\beta})}(d', \tau'(\varphi))$$

holds for some $d' \in D'$. Therefore, $P_{\Gamma^{\vdash_\beta}}(d', \varphi)$ also holds and hence $P_\Gamma(D \circ d', \varphi)$ holds, which means that $\varphi \in \Gamma^{\vdash_a}$.

(ii.c) β is a limit ordinal: straightforward. $\triangleleft$

4.3 Final remarks

In this chapter heterogeneous fibring was discussed and two solutions to the problem were presented. The first solution was based on consequence systems, observing that both deductive systems and semantics in a logic induce consequence systems. With consequence systems we can produce the fibring of logics presented, for instance, by a Hilbert calculus and by a sequent calculus. Moreover, we can define the fibring of logics where one of them is presented by a Hilbert calculus and the other by semantic structures.

The drawback of such an approach has to do with the fibring of logics presented by calculi. In this case we loose the notion of derivation. And we would like to be able to relate a derivation in the fibring with derivations in the component logics. The intuitive idea is that somehow a derivation in the fibring is composed by blocks where each block is a derivation in a component.

We introduced the notion of abstract proof system as the adequate setting for speaking about derivations. That is, in abstract proof systems, derivations are first-class citizens. This notion has for calculi the same role as ordered algebras had in Chapter 3 for logics presented by different semantics. In an abstract proof system we have a set of derivations with no structure. Hence, derivations can be, for instance, sequences of formulas or natural numbers. The nice thing is how we obtain derivations in the fibring, namely, by having an operator for combining derivations of the component logics.

We believe that the notion of fibring abstract proof systems can also be used to define homogeneous fibring of deductive systems if the objective is to emphasize the derivation structure. Some preliminary results appear in [69], where the fibring of sequent calculi is introduced not at the level of the rules but instead at the level of derivations. It is also worthwhile to note that some preservation results are easier to prove in this context, such as cut elimination.

A natural generalization of abstract proof systems would be what we can call abstract labeled proof systems. It is well known that many calculi are labeled. They are more flexible in the sense that they encode more logics than no labeled calculi, where one should provide a specific one for each logic. For instance, in modal logic there are several labeled calculi depending on the nature of the labels: worlds or truth-values. In the case of labeled calculi, we have to deal with heterogeneous calculi and also heterogeneous labels. The right setting for labels is still to be dealt with. Observe that the "universality" of labels is also recognized in network fibring as presented in Chapter 10.

We believe that heterogeneous fibring can also be investigated within the setting of multicategories and even polycategories following the ideas of Lambek and Szabo [176, 254] for proof systems. At first sight, it seems that the adequate notion of composition for polycategories has to be identified. Also some categorical constructions have to be studied as, for instance, fibrations. In this context, it will be possible to talk about logics that are substructural [229] such as for example linear logic. This opens the research topic of fibring substructural logics.

Chapter 5

Fibring non-truth functional logics

In this chapter we extend to non-truth functional logics the fibring techniques presented in Chapters 2 and 3. Some paraconsistent logics (in particular, some **LFI**s) constitute examples of non-truth functional logics. Paraconsistent logics were introduced in [73] and since then have been the object of continued attention, because of their theoretical and practical significance. In particular, the paraconsistent systems $\mathfrak{C}_n$ of [73] (see Example 2.2.9) are subsystems of propositional classical logic in which the principle of *Pseudo Scotus* $\gamma, (\neg\gamma) \vdash \varphi$ does not hold. It is well known that, in all the $\mathfrak{C}_n$ systems, negation cannot be given a truth functional semantics (see [209]).

From the proof-theoretical point of view, dealing with non-truth functional logic offers no particular difficulties. As before, in the fibring of two Hilbert calculi, the signature corresponds to the fibring of the signatures of the components and the inference rules are just the inference rules of both calculi.

From the semantic point of view, things are not so easy since we have to deal with (possibly) non-truth functional valuations. To establish the appropriate semantic setting, the main feature of the present approach is the use of a suitable auxiliary logic, called the metalogic, where the (possibly) non-truth functional valuations are defined. We consider conditional equational logic CEQ (see [132, 208]) as the metalogic, but we could have adopted, instead, any other classical metalogic where non-truth functional valuation semantics could be defined. Then, we extend the notion of fibring to this context and then we prove that this extended fibring preserves completeness under reasonable conditions.

The system $\mathfrak{C}_1^D$ of paraconsistent modal logic, introduced in [75], is then taken as an application example. It should be reasonable to expect that the mixed logic $\mathfrak{C}_1^D$ could be recovered by fibring the underlying modal and paraconsistent logics. However, the fibring of the two fragments (the modal with the paraconsistent), using the method we propose, produces a logic which is a little weaker than the original $\mathfrak{C}_1^D$. There is a simple explanation for this problem: $\mathfrak{C}_1^D$ contains an essential interaction axiom that cannot even be expressed in either of the logics being fibred. Taking this point into consideration, the paraconsistent modal logic $\mathfrak{C}_1^D$ can be easily recovered simply by adding the missing axiom to the obtained fibred logic, as well as a corresponding semantic restriction. This process is in line with the original idea of fibring as proposed in [108] (see Chapter 1).

The layout of this chapter is as follows: in Section 5.1, the notion of interpretation system presentation is introduced. The interpretation structures are also defined as being the models of the specifications. In Section 5.2 the notions of unconstrained and constrained fibring of interpretation system presentations is defined. In Section 5.3 we again use Hilbert calculus as the suitable proof-theoretic notion. In Section 5.4 some preservation results are established, namely, the preservation of soundness and the preservation of completeness. Section 5.5 discusses self-fibring in the context of non-truth functional logics. Finally, in Section 5.6 we present some final remarks.

The relevant material for this chapter is the work presented in [36].

5.1 Specifying valuation semantics

As we saw in Chapter 3, the definition of an algebraic semantics for a truth functional logic requires to endow it with models that are algebras (of truth values) over the signature of the logic and evaluate formulas by means of homomorphisms. However, some logics, called non-truth functional, do not accept this kind of semantical treatment. The approach taken in this chapter, first sketched in [61], is slightly different: we chose to work with two-sorted algebras of formulas and truth values, including the valuation map as an operation symbol between the two sorts. Thus, every model consist of a single (possibly non-truth functional) valuation map. As a particular case, truth functional logics appear by imposing the homomorphism conditions on the valuation maps. Technically, each model is a two-sorted algebra (of formulas and truth values) including a valuation operation satisfying some requirements, which are written in an appropriate conditional equational metalogic. As mentioned above, the metalogic adopted herein is CEQ (see [132, 208]).

As done in the previous chapters, we consider propositional based signatures of the form $C = \{C_k\}_{k\in\mathbb{N}}$ (see Chapter 2). In order to avoid confusion, such signatures are called *object signatures* along this chapter. This option is justified because we are also going to deal with the signatures for the metalogic, which are of a different nature. As previously done, we also consider as before the set of countable propositional symbols $\mathbb{P} \subseteq C_0$ and the set of schema variables Ξ. The language $L(C)$ is defined as before.

The next step is to define an equational signature induced by a given object signature C (see Definition 5.1.1 below). This equational signature can be seen as a kind of metalinguistic device which allows us to formally talk about the semantics of propositional logics based on signature C. It is thus convenient to consider two sorts: a sort ϕ (for formulas) and a sort τ (for truth values). As usual, if $\mathbf{S}$ is a set of sorts then $\mathbf{S}^*$ denotes the set of all strings (finite sequences) over $\mathbf{S}$; s^k will denote the string formed by k occurrences of sort s, and ϵ will denote the empty string. In the definition below, $O_{w\,s}$ will denote the set of operations with domain w and codomain s, for $w \in \mathbf{S}^*$ and $s \in \mathbf{S}$. For details about many-sorted algebras see, for instance, [133, 195]. Since this is enough for our purposes, we will only define the particular case of many-sorted first-order language to be used in this chapter. In Chapter 7, many-sorted higher-order languages will be treated in more detail.

Definition 5.1.1 Let C be a propositional signature, and let $\mathbf{S} = \{\phi, \tau\}$ be a set of sorts. The 2-sorted equational signature $\Sigma(C, \Xi) = \langle \mathbf{S}, O\rangle$ is the 2-sorted, first-order equational signature generated by the following sets of symbols:

- $O_{\epsilon\,\phi} = C_0 \cup \Xi$;
- $O_{\phi^k\,\phi} = C_k$, for $k > 0$;
- $O_{\phi\,\tau} = \{v\}$;
- $O_{\epsilon\,\tau} = \{\top, \bot\}$;
- $O_{\tau\,\tau} = \{-\}$;
- $O_{\tau^2\,\tau} = \{\sqcap, \sqcup, \sqsupset\}$;
- $O_{\omega\,s} = \emptyset$ in the other cases.

The signature $\Sigma(C, \Xi)$ is called the *induced metasignature*, and C is called the *object signature*. The sort ϕ is the sort for *formulas*, and the sort τ is the sort for *truth values*. We shall use $\Sigma(C)$ to denote the subsignature $\Sigma(C, \emptyset)$, that is, where $O_{\epsilon\,\phi} = C_0$. ∇

Recall that $O_{w\,s}$ is the set of function symbols from sort w to sort s. Thus, $O_{\phi^k\,\phi} = C_k$ means that each k-ary connective c of C is now considered as a function symbol $c : \phi^k \to \phi$. On the other hand, if $c \in C_0 \cup \Xi$ then c is considered as a function symbol $c : \epsilon \to \phi$, that is, a constant of sort ϕ. The truth values form an ordered algebra, and so there exist function symbols representing the basic operations. Thus, for instance, $\sqcap : \tau^2 \to \tau$ represents the (binary) infimum operator between truth values. As a consequence of this approach, the formulas of $L(C)$ are now terms of sort ϕ of the first-order language generated by $\Sigma(C, \Xi)$. Since $\Sigma(C, \Xi)$ is an equational signature, the symbols $\approx_\phi$ and $\approx_\tau$ for equality of sort ϕ and τ are taken for granted. From now on, we will omit the reference to the sort when using the equality predicates.

As mentioned above, the symbols $\top$, $\bot$, $-$, $\sqcap$, $\sqcup$ and $\sqsupset$ are used as generators of truth values. Additionally, the function symbol $v : \phi \to \tau$ represents a valuation map, taking formulas into truth values.

In order to construct the equational first-order language generated from $\Sigma(C,\Xi)$ we consider the set of variables $X_\phi = \{y_1, y_2, \dots\}$ and $X_\tau = \{x_1, x_2, \dots\}$ of sort ϕ and τ, respectively. For ease of notation we simply use X to denote the two-sorted family $\{X_\phi, X_\tau\}$.

Definition 5.1.2 Let $\Sigma(C,\Xi)$ be a 2-sorted equational signature.
The set of *non-truth functional terms of sort* ϕ is the set $T(C,\Xi)_\phi$ inductively defined as follows:

- $C_0 \cup \Xi \cup X_\phi \subseteq T(C,\Xi)_\phi$;
- $(c(t_1,\dots,t_k)) \in T(C,\Xi)_\phi$ whenever $k \geq 1$, $c \in C_k$ and $t_1,\dots,t_k \in T(C,\Xi)_\phi$.

The set of *non-truth functional terms of sort* τ is the set $T(C,\Xi)_\tau$ inductively defined as follows:

- $\{\top,\bot\} \cup X_\phi \subseteq T(C,\Xi)_\tau$;
- $v(t) \in T(C,\Xi)_\tau$ whenever $t \in T(C,\Xi)_\phi$;
- $-(t) \in T(C,\Xi)_\tau$ whenever $t \in T(C,\Xi)_\tau$;
- $c(t_1,t_2) \in T(C,\Xi)_\tau$ whenever $c \in \{\sqcap,\sqcup,\sqsupset\}$ and $t_1,t_2 \in T(C,\Xi)_\tau$. ∇

It is worth noting an important distinction: given the metasignature $\Sigma(C,\Xi)$, schema variables (that is, ξ_1,ξ_2, etc.) represent arbitrary formulas but only in the context of (propositional) Hilbert calculi defined over C. On the other hand, variables of sort ϕ (that is, y_1,y_2, etc.) represent arbitrary formulas in the metalanguage of CEQ. In this metalanguage, propositional schema variables appear as constants. For simplicity, we will use just "term" instead of "non-truth functional term".

Recall that a term t is called a closed term if it does not contain variables in X. We denote by $cT(C,\Xi)_\phi$ and $cT(C,\Xi)_\tau$ the sets of closed terms of sort ϕ and τ, respectively. Observe that $cT(C,\Xi)_\phi$ is the set of formulas $L(C)$. A *closed substitution* is a pair

$$\rho = \langle \rho_\phi, \rho_\tau \rangle$$

such that $\rho_\phi : X_\phi \to cT(C,\Xi)_\phi$ and $\rho_\tau : X_\tau \to cT(C,\Xi)_\tau$ are maps. From now on, for simplicity, we will write ρ to denote either ρ_ϕ or ρ_τ when no confusion arises.

The next step is to define valuation specifications (within the metalogic CEQ) over $\Sigma(C)$ and X. Recall that a CEQ-specification is a set of conditional equations of the general form:

$$(\text{equation}_1 \;\&\; \dots \;\&\; \text{equation}_n \;\longrightarrow\; \text{equation})$$

with $n \geq 0$. Each equation in the expression above is of the form $(t \approx t')$ such that t and t' are terms of the same sort built over $\Sigma(C)$ and X. It is important to notice that no schema variables occur in a conditional equation, by the very definition of $\Sigma(C)$. Of course, the sort of each equation is defined to be the sort of its terms. A conditional equation that only involves equations of a given sort is said to be a conditional equation of that sort. As usual in equational logic, conditional equations are universally quantified. For the sake of simplicity, we omit the quantifier, contrarily to the notation used in [208]. For instance,

$$(\longrightarrow (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2))))$$

is a conditional equation of sort τ, provided that $\wedge \in C_2$. From this example and the forthcoming ones, it should be clear that we only need to consider specifications containing exclusively conditional equations (or *meta-axioms*) of sort τ. This kind of specifications will be called τ-specifications from now on. The Hilbert calculus of CEQ (adapted from [208]) is a deductive system designed for deriving equations from a given set of conditional equations. That calculus consists of the usual rules for reflexivity, symmetry, transitivity and congruence of equality, plus an appropriate form of modus ponens. For the reader's convenience we briefly describe here the calculus CEQ which will allow us to prove equations from specifications. The inference rules of CEQ will be written as

$$\frac{P_1 \dots P_n}{\text{eq}}$$

where each premise P_i is either an equation eq_i or a conditional equation

$$(\text{eq}_1^i \ \& \ \dots \ \& \ \text{eq}_{n_i}^i \ \longrightarrow \ \text{eq}_i).$$

Definition 5.1.3 The Hilbert calculus of CEQ is composed by the following deduction rules:

$$\frac{}{(t \approx t)} \quad \text{reflexivity}$$

$$\frac{(t_1 \approx t_2)}{(t_2 \approx t_1)} \quad \text{symmetry}$$

$$\frac{(t_1 \approx t_2) \qquad (t_2 \approx t_3)}{(t_1 \approx t_3)} \quad \text{transitivity}$$

$$\frac{(t_{11} \approx t_{21}), \dots, (t_{1k} \approx t_{2k})}{(f(t_{11}, \dots, t_{1k}) \approx f(t_{21} \dots t_{2k}))} \quad \text{congruence}$$

$$\frac{((t_{11} \approx t_{21}) \ \& \ \dots \ \& \ (t_{1k} \approx t_{2k}) \longrightarrow (t_1 \approx t_2)), \quad (\rho(t_{11}) \approx \rho(t_{21})), \dots, (\rho(t_{1k}) \approx \rho(t_{2k}))}{(\rho(t_1) \approx \rho(t_2))} \quad \text{modus ponens}$$

In the congruence rule, f is a function symbol and in the modus ponens rule, ρ is a substitution. $\triangledown$

In the sequel we use $\vdash^{\text{CEQ}}_{\Sigma(C,\Xi)}$ to denote the corresponding consequence relation. Thus,

$$S \vdash^{\text{CEQ}}_{\Sigma(C,\Xi)} (t_1 \approx t_2)$$

means that there exists a derivation in CEQ of the equation $(t_1 \approx t_2)$ from the specification S.

Definition 5.1.4 An *interpretation system presentation* is a pair

$$\mathcal{S} = \langle C, S \rangle$$

where C is an object signature and S is a τ-specification over $\Sigma(C)$ (that is, a set of conditional equations of sort τ). ∇

Let $\mathcal{S} = \langle C, S \rangle$ be a given interpretation system presentation. As expected, a (semantical) interpretation structure for $\mathcal{S}$ is a first-order structure for the equational first-order language generated from $\Sigma(C, \Xi)$ satisfying the τ-specification S. More precisely (recall Definition 5.1.1):

Definition 5.1.5 Let $\Sigma(C, \Xi)$ be a signature. An *interpretation structure* over the signature $\Sigma(C, \Xi)$ is a triple

$$\mathcal{A} = \langle \mathcal{A}_\phi, \mathcal{A}_\tau, \cdot_\mathcal{A} \rangle$$

such that $\mathcal{A}_\phi$ and $\mathcal{A}_\tau$ are non-empty sets (the carrier of sorts ϕ and τ in $\mathcal{A}$, respectively) and $\cdot_\mathcal{A}$ is an interpretation map of the symbols in $\Sigma(C, \Xi)$ such that:

- $c_\mathcal{A} \in \mathcal{A}_\phi$ whenever $c \in C_0 \cup \Xi$;
- $c_\mathcal{A} : \mathcal{A}_\phi^k \to \mathcal{A}_\phi$ whenever $c \in C_k$, for $k > 0$;
- $v_\mathcal{A} : \mathcal{A}_\phi \to \mathcal{A}_\tau$;
- $\top_\mathcal{A}, \bot_\mathcal{A} \in \mathcal{A}_\tau$;
- $-_\mathcal{A} : \mathcal{A}_\tau \to \mathcal{A}_\tau$;
- $c_\mathcal{A} : \mathcal{A}_\tau^2 \to \mathcal{A}_\tau$ whenever $c \in \{\sqcap, \sqcup, \sqsupset\}$. ∇

From the definition above, it is clear that an interpretation structure $\mathcal{A}$ over $\Sigma(C, \Xi)$ is formed by two-algebras and a map between them as follows:

- an algebra $\mathcal{A}_\phi$ over C (of formulas);
- an algebra $\mathcal{A}_\tau$ over the signature of Heyting algebras (of truth values);
- a valuation map $v_\mathcal{A} : \mathcal{A}_\phi \to \mathcal{A}_\tau$ relating both algebras.

Of course, the interesting (or standard) cases are when $\mathcal{A}_\phi = L(C)$ and $\mathcal{A}_\tau$ is indeed a Heyting algebra with respect to their operations. In such cases, $v_\mathcal{A}$ is a valuation map into a Heyting algebra of truth values. As it will be shown in Proposition 5.1.20, as long as semantical entailment is concerned it is not necessary to restrict the interpretations to those such that $\mathcal{A}_\phi = L(C)$. With respect to the second requirement for an interpretation be 'standard', we consider the following definition. But before we present some notation. Let S be a τ-specification. We denote by $S^\bullet$ the τ-specification composed of the meta-axioms in S plus τ-equations over $\Sigma(C)$ specifying the class of all Heyting algebras (recall Example 3.1.4 in Chapter 3).

Definition 5.1.6 The class of interpretation structures presented by $\mathcal{S}$, denoted by

$$\text{INT}(\mathcal{S})$$

is the class of interpretations $\mathcal{A}$ over $\Sigma(C,\Xi)$ satisfying the specification $S^\bullet$. $\triangledown$

If $\mathcal{A} \in \text{INT}(\mathcal{S})$ then $\mathcal{A}_\tau$ is a Heyting algebra with respect to their operations. In particular, $\top_\mathcal{A}$ and $\bot_\mathcal{A}$ are the top and bottom elements of $\mathcal{A}_\tau$ with respect to the induced order. Observe that $\text{INT}(\mathcal{S})$ is always non-empty. Indeed, the trivial interpretation $\mathcal{A}$ with singleton carrier sets for both sorts ϕ and τ satisfies any set of conditional equations; in particular, $\mathcal{A}$ satisfies $S^\bullet$. Thus $\mathcal{A} \in \text{INT}(\mathcal{S})$.

Of course, Definition 5.1.6 lies on the rigorous notion of satisfaction of conditional equations. This notion coincides with the usual concept of satisfaction of formulas in first-order structures, which we briefly recall now.

Let $\mathcal{A}$ be an interpretation over $\Sigma(C,\Xi)$. An *assignment* over $\mathcal{A}$ is a pair

$$\alpha = \langle \alpha_\phi, \alpha_\tau \rangle$$

of mappings $\alpha_\phi : X_\phi \to \mathcal{A}_\phi$ and $\alpha_\tau : X_\tau \to \mathcal{A}_\tau$. Analogously to what was done in Definition 3.1.12 of Chapter 3, the denotation of a metaterm t given an assignment α can be defined:

Definition 5.1.7 Given an interpretation $\mathcal{A}$ over $\Sigma(C,\Xi)$ and an assignment α over $\mathcal{A}$, the denotation map

$$[\![\cdot]\!]^\alpha_\mathcal{A} : T(C,\Xi)_\phi \cup T(C,\Xi)_\tau \to \mathcal{A}_\phi \cup \mathcal{A}_\tau$$

is defined as usual:

- $[\![t]\!]^\alpha_\mathcal{A} = t_\mathcal{A}$ if $t \in C_0 \cup \Xi \cup \{\top, \bot\}$;
- $[\![t]\!]^\alpha_\mathcal{A} = \alpha_s(t)$ if $t \in X_s$ for some $s \in \{\phi, \tau\}$;
- $[\![(c(t_1,\ldots,t_k))]\!]^\alpha_\mathcal{A} = c_\mathcal{A}([\![t_1]\!]^\alpha_\mathcal{A},\ldots,[\![t_k]\!]^\alpha_\mathcal{A})$ whenever $k \geq 1$, $t_1,\ldots,t_k \in T(C,\Xi)_\phi$ and $c \in C_k$;

- $[\![v(t)]\!]^{\alpha}_{\mathcal{A}} = v_{\mathcal{A}}([\![t]\!]^{\alpha}_{\mathcal{A}})$ whenever $t \in T(C,\Xi)_{\phi}$;
- $[\![-(t)]\!]^{\alpha}_{\mathcal{A}} = -_{\mathcal{A}}([\![t]\!]^{\alpha}_{\mathcal{A}})$ whenever $t \in T(C,\Xi)_{\tau}$;
- $[\![c(t_1,t_2)]\!]^{\alpha}_{\mathcal{A}} = c_{\mathcal{A}}([\![t_1]\!]^{\alpha}_{\mathcal{A}}, [\![t_2]\!]^{\alpha}_{\mathcal{A}})$ whenever $c \in \{\sqcap, \sqcup, \sqsupset\}$ and $t_1, t_2 \in T(C,\Xi)_{\tau}$. ∇

In the case of a closed term t we just write $[\![t]\!]_{\mathcal{A}}$ for its denotation in $\mathcal{A}$.

It is worth noting that, if t is a term of sort s, then $[\![t]\!]^{\alpha}_{\mathcal{A}} \in \mathcal{A}_s$, for $s \in \{\phi, \tau\}$.

Given an interpretation $\mathcal{A}$ over $\Sigma(C,\Xi)$, we say that $\mathcal{A}$ *satisfies* a conditional equation

$$((t_{11} \approx t_{21}) \ \& \ \dots \ \& \ (t_{1k} \approx t_{2k}) \longrightarrow (t_1 \approx t_2))$$

if, for every assignment α over $\mathcal{A}$, if $[\![t_{1i}]\!]^{\alpha}_{\mathcal{A}} = [\![t_{2i}]\!]^{\alpha}_{\mathcal{A}}$ for every $i = 1, \dots, k$ then $[\![t_1]\!]^{\alpha}_{\mathcal{A}} = [\![t_2]\!]^{\alpha}_{\mathcal{A}}$. In particular, $\mathcal{A}$ satisfies an equation

$$(\longrightarrow (t_1 \approx t_2))$$

whenever $[\![t_1]\!]^{\alpha}_{\mathcal{A}} = [\![t_2]\!]^{\alpha}_{\mathcal{A}}$ for every assignment α over $\mathcal{A}$.

Now it should be clear that, as mentioned above, any interpretation structure $\mathcal{A}$ presented by $\mathcal{S}$ is such that $\mathcal{A}_{\tau}$ is a Heyting algebra.

For simplicity, we introduce the following abbreviations on the set $T(C,\Xi)_{\tau}$:

- $\equiv (t_1, t_2) =_{\text{def}} \sqcap(\sqsupset (t_1,t_2), \sqsupset (t_2,t_1))$;
- $(t_1 \leq t_2) =_{\text{def}} (\sqcap(t_1,t_2) \approx t_1)$.

Clearly, the relation symbol $\leq$ denotes a partial order on truth values. Furthermore, the partial order is a bounded lattice with meet $\sqcap$, join $\sqcup$, top $\top$ and bottom $\bot$ (see [21]).

Accordingly to the notation above introduced, given an interpretation structure $\mathcal{A}$ and $a_1, a_2 \in \mathcal{A}_{\tau}$,

$$a_1 \leq_{\mathcal{A}} a_2 \text{ and } \equiv_{\mathcal{A}} (a_1, a_2)$$

are abbreviations of $\sqcap_{\mathcal{A}}(a_1,a_2) = a_1$ and $\sqcap_{\mathcal{A}}(\sqsupset_{\mathcal{A}} (a_1,a_2), \sqsupset_{\mathcal{A}} (a_2,a_1))$, respectively.

It is also well known that the Heyting algebra axioms further entail the following result:

Proposition 5.1.8 *Let $\mathcal{S}$ be an interpretation system presentation, t_1 and t_2 terms of sort τ and $\mathcal{A} \in \text{INT}(\mathcal{S})$. Then, for every assignment α over $\mathcal{A}$:*

$$[\![t_1]\!]^{\alpha}_{\mathcal{A}} \leq_{\mathcal{A}} [\![t_2]\!]^{\alpha}_{\mathcal{A}} \quad \textit{if and only if} \quad \sqsupset_{\mathcal{A}} ([\![t_1]\!]^{\alpha}_{\mathcal{A}}, [\![t_2]\!]^{\alpha}_{\mathcal{A}}) = \top_{\mathcal{A}}$$

and

$$[\![t_1]\!]^{\alpha}_{\mathcal{A}} = [\![t_2]\!]^{\alpha}_{\mathcal{A}} \quad \textit{if and only if} \quad \equiv_{\mathcal{A}} ([\![t_1]\!]^{\alpha}_{\mathcal{A}}, [\![t_2]\!]^{\alpha}_{\mathcal{A}}) = \top_{\mathcal{A}}.$$

The notion of interpretation system presentation can be illustrated with some examples that will be used throughout the rest of the chapter.

Example 5.1.9 Paraconsistent Hilbert calculus $\mathfrak{C}_1$ (recall Example 2.2.9):

- Object signature C:
  - $C_0 = \mathbb{P} \cup \{\mathbf{t}, \mathbf{f}\}$;
  - $C_1 = \{\neg\}$;
  - $C_2 = \{\wedge, \vee, \Rightarrow\}$.
- Meta-axioms S:
  - Truth-values axioms: further axioms in order to obtain a specification of the class of all Boolean algebras through $S^\bullet$ (recalling that $S^\bullet$ always contains the axioms of Heyting algebras). As it is well-known, it is enough to add the following equation:

    $$(\ \longrightarrow (-(-(x_1)) \approx x_1)).$$
  - Valuation axioms:

    $$\begin{array}{l}
(\ \longrightarrow (v(\mathbf{t}) \approx \top))\\
(\ \longrightarrow (v(\mathbf{f}) \approx \bot))\\
(\ \longrightarrow (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2))))\\
(\ \longrightarrow (v(y_1 \vee y_2) \approx \sqcup(v(y_1), v(y_2))))\\
(\ \longrightarrow (v(y_1 \Rightarrow y_2) \approx \sqsupset(v(y_1), v(y_2))))\\
(\ \longrightarrow (v(\neg(\neg\, y_1)) \leq v(y_1)))\\
(\ \longrightarrow (v((\neg\, y_1) \wedge y_1^\circ) \approx -(v(y_1))))\\
(\ \longrightarrow \sqcap(v(y_1^\circ), v(y_2^\circ)) \leq v((y_1 \wedge y_2)^\circ))\\
(\ \longrightarrow \sqcap(v(y_1^\circ), v(y_2^\circ)) \leq v((y_1 \vee y_2)^\circ))\\
(\ \longrightarrow \sqcap(v(y_1^\circ), v(y_2^\circ)) \leq v((y_1 \Rightarrow y_2)^\circ)).
\end{array}$$

As usual in the $\mathfrak{C}_n$ systems, γ° is an abbreviation of $(\neg(\gamma \wedge (\neg\,\gamma)))$. The semantics above was originally introduced in [74], but defined over the Boolean algebra *2*, that is, as a bivaluation semantics.

It is worth noting that herein we are using Boolean algebras as a metamathematical environment sufficient to carry out the computations of truth values for the formulas in $\mathfrak{C}_1$. In particular, we are not introducing any unary operator in the Boolean algebras corresponding to paraconsistent negation, but we are computing the values of formulas of the form $(\neg\,\gamma)$ by means of conditional equations in the algebras. In other words, $\neg$ does not correspond to the Boolean algebra complement $-$. Therefore we are not attempting to algebraize $\mathfrak{C}_1$ in any usual way. The question of algebraizing paraconsistent logic is a separate issue and we refer the interested reader to [209, 180, 30, 29].

Observe that every bivaluation introduced in [74] has a counterpart in $\textsc{Int}(\mathcal{S})$. Furthermore, the additional interpretation structures (that is, those defined over other Boolean algebras than $\mathcal{2}$) do not change the semantic entailment to be defined in Definition 5.1.15 below. It is easy to extend this example to the whole hierarchy $\mathfrak{C}_n$ by specifying appropriately the paraconsistent n-valuations introduced in [182].
$\triangledown$

Example 5.1.10 The logics of formal inconsistency (**LFI**s) were mentioned in Example 1.4.6 of Chapter 1. These logics, introduced in [51] (see also [49]) offer a new perspective for paraconsistent logics. The basic idea consists of considering a (primitive or defined) *consistency* connective $\circ$ in order to control *explosiveness* of contradictions. Thus, in general

$$\varphi, (\neg\varphi) \nvdash \psi$$

for some φ and ψ. However,

$$(\circ\varphi), \varphi, (\neg\varphi) \vdash \psi$$

for every φ and ψ. That is, a contradiction *plus* the consistency of the contradictory formula defines a trivial theory. The logic $\mathfrak{C}_1$ described above is an example of an **LFI**, in which $(\circ\varphi)$ is given by the formula $(\neg(\varphi \wedge (\neg\varphi)))$. It is possible to use an *inconsistency* connective $\bullet$ instead of a consistency connective, or use both of them.

Some **LFI**s such as P^1 (introduced in [244]) and J_3 (introduced as a modal paraconsistent logic in [85] and re-introduced in [89] as an **LFI**) are truth functional, because they have a matrix semantics. However, most **LFI**s (and paraconsistent logics, in general) have a non-truth functional semantics: the consistency (or inconsistency) operator, as well as the negation, are not, in general, truth functional connectives (see Definition 5.1.12 below). As a consequence of this, these logics are not *congruential*: if φ_i is inter-derivable with ψ_i then, in general, $\gamma(\varphi_1, \ldots, \varphi_n)$ is not inter-derivable with $\gamma(\psi_1, \ldots, \psi_n)$ for some formulas $\gamma(p_1, \ldots, p_n)$, φ_i and ψ_i (for $i = 1, \ldots, n$).

Frequently, an adequate (non-truth functional) bivaluation semantics can be given for such logics. As it was done for $\mathfrak{C}_1$ in Example 5.1.9, this bivaluation semantics can be revamped in our setting, by changing the Boolean algebra $\mathcal{2}$ for an arbitrary Boolean algebra. For instance, the logic **mbC**, a basic **LFI** (see [51]), admits the following presentation:

- Object signature C:
  - $C_0 = \mathbb{P} \cup \{\mathbf{t}, \mathbf{f}\}$;
  - $C_1 = \{\neg, \circ\}$;
  - $C_2 = \{\wedge, \vee, \Rightarrow\}$.
- Meta-axioms S:

– Truth-values axioms: idem to Example 5.1.9;

– Valuation axioms:

$(\;\longrightarrow (v(\mathbf{t}) \approx \top))$
$(\;\longrightarrow (v(\mathbf{f}) \approx \bot))$
$(\;\longrightarrow (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2))))$
$(\;\longrightarrow (v(y_1 \vee y_2) \approx \sqcup(v(y_1), v(y_2))))$
$(\;\longrightarrow (v(y_1 \Rightarrow y_2) \approx \sqsupset(v(y_1), v(y_2))))$
$(\;\longrightarrow (v((\neg\, y_1) \wedge (\circ y_1)) \approx -(v(y_1)))).$

As in Example 5.1.9, every paraconsistent bivaluation introduced in [49] occurs in $\textsc{Int}(\mathcal{S})$, and the interpretation structures defined over Boolean algebras other than $\mathit{2}$ do not change the semantic entailment. The logic **Ci**, an **LFI** obtained from **mbC** by adding the axiom schemata $((\neg(\neg\xi))\Rightarrow\xi)$ and $((\neg(\circ\xi))\Rightarrow(\xi\wedge(\neg\xi)))$ (recall Example 1.4.6), can be characterized semantically by adding the following clauses to S (adapting the bivaluation semantics given in [49]):

$(\;\longrightarrow (v(\neg(\neg\, y_1)) \leq v(y_1)))$

$(\;\longrightarrow (v(\neg(\circ y_1)) \leq v(y_1 \wedge (\neg\, y_1)))).$ ▽

Remark 5.1.11 At this point its is interesting to observe how does the interpretation work. For instance, the equation

$$(\;\longrightarrow (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2))))$$

states the following: for every interpretation structure $\mathcal{A}$ and every assignment α, $v_{\mathcal{A}} : \mathcal{A}_\phi \to \mathcal{A}_\tau$ must be a map such that

$$v_{\mathcal{A}}([\![(y_1 \wedge y_2)]\!]^\alpha_{\mathcal{A}}) = \sqcap_{\mathcal{A}}(v_{\mathcal{A}}([\![y_2]\!]^\alpha_{\mathcal{A}}), v_{\mathcal{A}}([\![y_2]\!]^\alpha_{\mathcal{A}}))).$$

Therefore,

$$v_{\mathcal{A}}([\![\varphi]\!]_{\mathcal{A}} \wedge_{\mathcal{A}} [\![\psi]\!]_{\mathcal{A}}) = \sqcap_{\mathcal{A}}(v_{\mathcal{A}}([\![\varphi]\!]_{\mathcal{A}}), v_{\mathcal{A}}([\![\psi]\!]_{\mathcal{A}})))$$

for every $\varphi, \psi \in L(C)$. As observed above, this condition is slightly more general than consider valuation maps $v_{\mathcal{A}} : L(C) \to \mathcal{A}_\tau$ such that

$$v_{\mathcal{A}}(\varphi \wedge \psi) = \sqcap_{\mathcal{A}}(v_{\mathcal{A}}(\varphi), v_{\mathcal{A}}(\psi)))$$

for every $\varphi, \psi \in L(C)$, because the use of the "intermediate" algebra of formulas $\mathcal{A}_\phi$. As we shall see in Proposition 5.1.20 below, this apparent generalization has no effects in the notion of entailment. ▽

Now we are able to give a rigorous definition of what we mean by a non-truth functional connective. From this, it is possible to formulate the meaning of non-truth functional semantics in our context. In order to be as general as possible, we will consider, besides the primitive connectives given in the signature, the derived ones. In rigorous terms, a *derived connective* of arity k is a λ-term $\lambda y_1 \dots y_k \,.\, \varphi$ such that the variables occurring in the schema formula φ are included in the set $\{y_1, \dots, y_k\}$. Clearly, any connective $c \in C_k$ is also a derived connective: it is enough to consider the λ-term $\lambda y_1 \dots y_k \,.\, (c(y_1, \dots, y_k))$.

Definition 5.1.12 A derived connective $\lambda y_1 \dots y_k \,.\, \varphi$ is said to be *truth functional* in a given interpretation system presentation $\mathcal{S}$ if

$$S^\bullet \vdash^{\text{CEQ}}_{\Sigma(C,\Xi)} (v(\varphi) \approx \rho^{v(y)}_{\ x}(t))$$

for some τ-term t written only on the variables $x_1, \dots, x_k$, where $\rho^{v(y)}_{\ x}$ is the substitution such that $\rho^{v(y)}_{\ x}(x_n) = v(y_n)$ for every $n \geq 1$.

If it is not possible to fulfill the above requirement, the connective is said to be *non-truth functional* in $\mathcal{S}$. ▽

It should be clear that showing that a certain connective is non-truth functional can be a very hard task.

Example 5.1.13 Consider paraconsistent logic $\mathfrak{C}_1$. Then a new negation can be defined as the derived connective $\sim \; =: \lambda y_1 \,.\, ((\neg\, y_1) \wedge y_1^\circ)$. Of course $\sim$ is truth functional: it is enough to take t as $-(x_1)$. From this it follows that $\sim$ is in fact a classical negation.

The classical equivalence is also definable in $\mathfrak{C}_1$ as

$$\Leftrightarrow \; := \lambda y_1 y_2 \,.\, ((y_1 \Rightarrow y_2) \wedge (y_2 \Rightarrow y_1)).$$

As expected, $\Leftrightarrow$ is truth functional: take t as $\equiv (x_1, x_2)$.

The primitive conjunction $\lambda y_1 y_2 \,.\, (y_1 \wedge y_2)$, disjunction $\lambda y_1 y_2 \,.\, (y_1 \vee y_2)$, and implication $\lambda y_1 y_2 \,.\, (y_1 \Rightarrow y_2)$ are also truth functional, as the reader can easily check.

On the other hand, as proven in [209], the paraconsistent negation $\lambda y_1 \,.\, (\neg\, y_1)$ is non-truth functional. ▽

Now we describe in our setting a logic system which will be used throughout the rest of this chapter as a case-study.

Example 5.1.14 Modal calculus **KD**:

- Object signature C:
  - $C_0 = \mathbb{P} \cup \{\mathbf{t}, \mathbf{f}\}$;
  - $C_1 = \{\neg, \Box\}$;
  - $C_2 = \{\wedge, \vee, \Rightarrow\}$.

- Meta-axioms S:
  - Truth values axioms:

    Further axioms in order to obtain a specification of the class of all Boolean algebras, as it was done in the previous example.
  - Valuation axioms:

$$
\begin{aligned}
&(\ \longrightarrow (v(\mathbf{t}) \approx \top))\\
&(\ \longrightarrow (v(\mathbf{f}) \approx \bot))\\
&(\ \longrightarrow (v(\neg\, y_1) \approx -(v(y_1))))\\
&(\ \longrightarrow (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2))))\\
&(\ \longrightarrow (v(y_1 \vee y_2) \approx \sqcup(v(y_1), v(y_2))))\\
&(\ \longrightarrow (v(y_1 \Rightarrow y_2) \approx \sqsupset(v(y_1), v(y_2))))\\
&(\ \longrightarrow (v(\Box\, \mathbf{t}) \approx \top))\\
&(\ \longrightarrow (v(\Box(y_1 \wedge y_2)) \approx \sqcap(v(\Box\, y_1), v(\Box\, y_2))))\\
&(\ \longrightarrow (\sqcap(v(\Box\, y_1), v(\neg(\Box(\neg\, y_1)))) \approx v(\Box\, y_1))\\
&((v(y_1) \approx v(y_2)) \longrightarrow (v(\Box\, y_1) \approx v(\Box\, y_2))).
\end{aligned}
$$

It is worth noting that every interpretation structure over C induced by a Kripke structure (recall Example 3.1.7 in Chapter 3) has a counterpart in $\textsc{Int}(\mathcal{S})$: it is enough to consider the Boolean algebra of truth values given by the power set $\wp W$ of the set of worlds W. As in Example 5.1.9, the extra interpretation structures (that is, those defined over Boolean algebras different from $\wp W$) do not change the semantic entailment to be introduced in Definition 5.1.15. $\triangledown$

In the interpretation system presentation above, all the derived connectives (with exception of $\Box$) are truth functional. In order to consider the modality $\lambda y_1 \,.\, \Box\, y_1$ as truth functional, we would need an extra generator L in $O_{\tau\,\tau}$ such that:

$$
\begin{aligned}
&(\ \longrightarrow (L(\top) \approx \top))\\
&(\ \longrightarrow (L(\sqcap(x_1, x_2)) \approx \sqcap(L(x_1), L(x_2))))\\
&(\ \longrightarrow (\sqcap(L(x_1), -(L(-(x_1))) \approx L(x_1))))\\
&(\ \longrightarrow (v(\Box\, y_1) \approx L(v(y_1)))).
\end{aligned}
$$

Note that these axioms on L are very closely related to the last four valuation axioms used in Example 5.1.14. Of course such an operation L could be defined over the set of truth values according to the axioms above, in order to specify the modal algebras described in Example 3.1.8 of Chapter 3. However, this move would violate our definition of interpretation structures, because of the inclusion of an extra operator. This explains the last four axioms given in Example 5.1.14, which allowed us to specify the intended class of modal algebras without using the extra operation L.

The notion of (global and local) semantic entailments in the present setting can be now defined.

Definition 5.1.15 Given an interpretation system presentation $\mathcal{S}$, a set $\Gamma \subseteq L(C)$ and a formula $\varphi \in L(C)$, we say that:

- $\Gamma \vDash^g_{\mathcal{S}} \varphi$ (Γ *globally entails* φ) if, for every $\mathcal{A} \in \textsc{Int}(\mathcal{S})$, $v_{\mathcal{A}}([\![\gamma]\!]_{\mathcal{A}}) = \top_{\mathcal{A}}$ for each $\gamma \in \Gamma$ implies $v_{\mathcal{A}}([\![\varphi]\!]_{\mathcal{A}}) = \top_{\mathcal{A}}$;
- $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$ (Γ *locally entails* φ) if, for every $\mathcal{A} \in \textsc{Int}(\mathcal{S})$ and every $b \in \mathcal{A}_\phi$, $v_{\mathcal{A}}(b) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\gamma]\!]_{\mathcal{A}})$ for each $\gamma \in \Gamma$ implies $v_{\mathcal{A}}(b) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\varphi]\!]_{\mathcal{A}})$. $\triangledown$

Observe that $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$ implies $\Gamma \vDash^g_{\mathcal{S}} \varphi$ provided that for every $\mathcal{A} \in \textsc{Int}(\mathcal{S})$ there exists $b \in \mathcal{A}_\phi$ such that $v_{\mathcal{A}}(b) = \top_{\mathcal{A}}$. On the other hand, $\Gamma \vDash^g_{\mathcal{S}} \varphi$ implies $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$, provided that $\Gamma = \emptyset$.

The reader should note the analogy between the semantic definitions introduced above and the corresponding notions in Definition 3.3.1.

As it was observed above, we would expect to consider interpretation structures $\mathcal{A}$ such that $\mathcal{A}_\phi = L(C)$, and then the valuation map $v_{\mathcal{A}}$ should be a "real" valuation $v_{\mathcal{A}} : L(C) \to \mathcal{A}_\tau$. We will prove in Proposition 5.1.20 below that nothing changes with the use of generalized valuation maps as long as entailment is concerned.

Definition 5.1.16 Let $\mathcal{A} \in \textsc{Int}(\mathcal{S})$. We say that $\mathcal{A}$ is a *standard* interpretation structure presented by $\mathcal{S}$ if $\mathcal{A}_\phi = L(C)$ and $\cdot_{\mathcal{A}}$ is the identity map over the symbols of C. $\triangledown$

We denote by $\textsc{Int}_{st}(\mathcal{S})$ the class of all standard interpretation structures.

From the definition above, a standard interpretation $\mathcal{A}$ determines a real valuation map $v_{\mathcal{A}} : L(C) \to \mathcal{A}_\tau$. Observe that if $\mathcal{A}$ is standard then

$$[\![\varphi]\!]_{\mathcal{A}} = \sigma_{\mathcal{A}}(\varphi)$$

where $\sigma_{\mathcal{A}} : \Xi \to L(C)$ is the substitution such that $\sigma_{\mathcal{A}}(\xi) = [\![\xi]\!]_{\mathcal{A}}$ for every $\xi \in \Xi$. In particular, $[\![\varphi]\!]_{\mathcal{A}} = \varphi$ if φ is a ground formula, that is, a formula without schema variables. These formulas are the interesting ones in what concerns to valuations, since ground formulas are the genuine formulas.

Now we introduce some useful technical definitions and results. Recall that $\Xi = \{\xi_n : n \in \mathbb{N}\}$.

Definition 5.1.17 The substitutions σ^+ and σ^- are as follows:

- $\sigma^+ : \Xi \to L(C)$ is given by $\sigma^+(\xi_i) = \xi_{i+1}$ for every $i \geq 1$;
- $\sigma^- : \Xi \to L(C)$ is given by $\sigma^-(\xi_1) = \xi_1$ and $\sigma^-(\xi_i) = \xi_{i-1}$ for every $i \geq 2$. $\triangledown$

Note that $\sigma^+(\varphi)$ is a variant of φ where ξ_1 does not occur. Furthermore, it is easy to prove that $\sigma^-(\sigma^+(\varphi)) = \varphi$.

Given an interpretation system presentation $\mathcal{S}$, we say that φ *is entailed by* Γ, denoted by

$$\Gamma \mathrel{\widehat{\models^{\ell}_{\mathcal{S}}}} \varphi$$

if, for every $\mathcal{A} \in \text{INT}(\mathcal{S})$, $v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\gamma)]\!]_{\mathcal{A}})$ for each $\gamma \in \Gamma$ implies $v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\varphi)]\!]_{\mathcal{A}})$.

Proposition 5.1.18 *Let $\mathcal{S}$ be an interpretation system presentation, and consider $\Gamma \cup \{\varphi\} \subseteq L(C)$. Then:*

$$\Gamma \models^{\ell}_{\mathcal{S}} \varphi \text{ if and only if } \Gamma \mathrel{\widehat{\models^{\ell}_{\mathcal{S}}}} \varphi.$$

Proof. Suppose that $\Gamma \models^{\ell}_{\mathcal{S}} \varphi$ and let $\mathcal{A} \in \text{INT}(\mathcal{S})$ such that

$$v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\gamma)]\!]_{\mathcal{A}})$$

for each $\gamma \in \Gamma$. Consider the interpretation structure $\mathcal{A}^+$ such that $\mathcal{A}^+_{\phi} = \mathcal{A}_{\phi}$, $\mathcal{A}^+_{\tau} = \mathcal{A}_{\tau}$, $\xi_{\mathcal{A}^+} = \sigma^+(\xi)_{\mathcal{A}}$ for every $\xi \in \Xi$ and $\cdot_{\mathcal{A}^+}$ coincides with $\cdot_{\mathcal{A}}$ on the other symbols. Since no schema variables occur in $S^{\bullet}$ then $\mathcal{A}^+ \in \text{INT}(\mathcal{S})$. It is easy to prove that

$$[\![\psi]\!]_{\mathcal{A}^+} = [\![\sigma^+(\psi)]\!]_{\mathcal{A}}$$

for every $\psi \in L(C)$. Let $b = [\![\xi_1]\!]_{\mathcal{A}}$. Since $v_{\mathcal{A}^+}(b) \leq_{\mathcal{A}^+} v_{\mathcal{A}^+}([\![\gamma]\!]_{\mathcal{A}^+})$ for each $\gamma \in \Gamma$ then $v_{\mathcal{A}^+}(b) \leq_{\mathcal{A}^+} v_{\mathcal{A}^+}([\![\varphi]\!]_{\mathcal{A}^+})$ and so $v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\varphi)]\!]_{\mathcal{A}})$. This proves that $\Gamma \mathrel{\widehat{\models^{\ell}_{\mathcal{S}}}} \varphi$.

Conversely, suppose that $\Gamma \mathrel{\widehat{\models^{\ell}_{\mathcal{S}}}} \varphi$ and let $\mathcal{A} \in \text{INT}(\mathcal{S})$ and $b \in \mathcal{A}_{\phi}$ such that $v_{\mathcal{A}}(b) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\gamma]\!]_{\mathcal{A}})$ for each $\gamma \in \Gamma$. Consider the interpretation structure $\mathcal{A}^-$ such that $\mathcal{A}^-_{\phi} = \mathcal{A}_{\phi}$, $\mathcal{A}^-_{\tau} = \mathcal{A}_{\tau}$, $\xi_{1\mathcal{A}^-} = b$, $\xi_{n\mathcal{A}^-} = \sigma^-(\xi_n)_{\mathcal{A}}$ for every $n \geq 2$, and the mapping $\cdot_{\mathcal{A}^-}$ coincides with $\cdot_{\mathcal{A}}$ on the other symbols. Then

$$[\![\sigma^+(\psi)]\!]_{\mathcal{A}^-} = [\![\psi]\!]_{\mathcal{A}}$$

for every $\psi \in L(C)$. No schema variables occur in $S^{\bullet}$, then $\mathcal{A}^- \in \text{INT}(\mathcal{S})$. From this the result follows easily. ◁

For o $\in \{g, \ell\}$, consider the relation $\overline{\models^{o}_{\mathcal{S}}}$ obtained from the corresponding relation $\models^{o}_{\mathcal{S}}$ of Definition 5.1.15 by taking $\text{INT}_{st}(\mathcal{S})$ instead of $\text{INT}(\mathcal{S})$. Then the following result holds:

Corollary 5.1.19 *Let $\mathcal{S}$ be an interpretation system presentation, and consider $\Gamma \cup \{\varphi\} \subseteq L(C)$. Then, $\Gamma \mathrel{\overline{\models^{o}_{\mathcal{S}}}} \varphi$ if and only if, for every $\mathcal{A} \in \text{INT}_{st}(\mathcal{S})$,*

$$v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\gamma)]\!]_{\mathcal{A}})$$

for each $\gamma \in \Gamma$ implies

$$v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\varphi)]\!]_{\mathcal{A}}).$$

Proof. The proof is identical to that of Proposition 5.1.18, using the definitions above: it is enough to notice that, if $\mathcal{A} \in \text{INT}_{st}(\mathcal{S})$ then both $\mathcal{A}^+$ and $\mathcal{A}^-$ are in $\text{INT}_{st}(\mathcal{S})$. ◁

Proposition 5.1.20 *For every interpretation system presentation* $\mathcal{S}$, *every set* $\Gamma \cup \{\varphi\} \subseteq L(C)$ *and* $o \in \{g, \ell\}$,

$$\Gamma \overline{\vDash^o_{\mathcal{S}}} \varphi \text{ if and only if } \Gamma \vDash^o_{\mathcal{S}} \varphi.$$

Proof.
Part 1: Global entailment.

Let $\mathcal{S}$ be an interpretation system presentation and let $\Gamma \cup \{\varphi\} \subseteq L(C)$. Assume that $\Gamma \overline{\vDash^g_{\mathcal{S}}} \varphi$, and suppose that $\mathcal{A} \in \text{INT}(\mathcal{S})$ is such that $v_{\mathcal{A}}([\![\gamma]\!]_{\mathcal{A}}) = \top_{\mathcal{A}}$ for each $\gamma \in \Gamma$. Consider the interpretation structure $\bar{\mathcal{A}}$ such that:

- $\bar{\mathcal{A}}_\phi = L(C)$ and $\bar{\mathcal{A}}_\tau = \mathcal{A}_\tau$;
- $\cdot_{\bar{\mathcal{A}}}$ is the identity map on the symbols of C and Ξ;
- $v_{\bar{\mathcal{A}}} = v_{\mathcal{A}} \circ ([\![\cdot]\!]_{\mathcal{A}}|_{L(C)})$;
- $\cdot_{\bar{\mathcal{A}}}$ coincides with $\cdot_{\mathcal{A}}$ on the other symbols.

It is clear that $\bar{\mathcal{A}} \in \text{INT}_{st}(\mathcal{S})$ such that $[\![\psi]\!]_{\bar{\mathcal{A}}} = \psi$ and $v_{\bar{\mathcal{A}}}([\![\psi]\!]_{\bar{\mathcal{A}}}) = v_{\mathcal{A}}([\![\psi]\!]_{\mathcal{A}})$ for every $\psi \in L(C)$. Moreover, $\top_{\bar{\mathcal{A}}} = \top_{\mathcal{A}}$. Thus,

$$v_{\bar{\mathcal{A}}}([\![\gamma]\!]_{\bar{\mathcal{A}}}) = \top_{\bar{\mathcal{A}}}$$

for each $\gamma \in \Gamma$ and so $v_{\bar{\mathcal{A}}}([\![\varphi]\!]_{\bar{\mathcal{A}}}) = \top_{\bar{\mathcal{A}}}$. Therefore $v_{\mathcal{A}}([\![\varphi]\!]_{\mathcal{A}}) = \top_{\mathcal{A}}$. This shows that $\Gamma \vDash^g_{\mathcal{S}} \varphi$.

The converse is obvious, because $\text{INT}_{st}(\mathcal{S}) \subseteq \text{INT}(\mathcal{S})$.

Part 2: Local entailment.

Let $\mathcal{S}$ be an interpretation system presentation and let $\Gamma \cup \{\varphi\} \subseteq L(C)$. Assume that $\Gamma \overline{\vDash^\ell_{\mathcal{S}}} \varphi$, and suppose that $\mathcal{A} \in \text{INT}(\mathcal{S})$ is such that

$$v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\gamma)]\!]_{\mathcal{A}})$$

for each $\gamma \in \Gamma$. Consider $\bar{\mathcal{A}} \in \text{INT}_{st}(\mathcal{S})$ defined as in Part 1. Then

$$v_{\bar{\mathcal{A}}}([\![\xi_1]\!]_{\bar{\mathcal{A}}}) \leq_{\bar{\mathcal{A}}} v_{\bar{\mathcal{A}}}([\![\sigma^+(\gamma)]\!]_{\bar{\mathcal{A}}})$$

for each $\gamma \in \Gamma$ and so $v_{\bar{\mathcal{A}}}([\![\xi_1]\!]_{\bar{\mathcal{A}}}) \leq_{\bar{\mathcal{A}}} v_{\bar{\mathcal{A}}}([\![\sigma^+(\varphi)]\!]_{\bar{\mathcal{A}}})$, by Corollary 5.1.19. Thus

$$v_{\mathcal{A}}([\![\xi_1]\!]_{\mathcal{A}}) \leq_{\mathcal{A}} v_{\mathcal{A}}([\![\sigma^+(\varphi)]\!]_{\mathcal{A}}).$$

This means that $\Gamma \widehat{\vDash^\ell_{\mathcal{S}}} \varphi$ and so, by Proposition 5.1.18, $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$. The converse follows easily, since $\text{INT}_{st}(\mathcal{S}) \subseteq \text{INT}(\mathcal{S})$. ◁

From the last result it is clear that, in terms of entailment, nothing changes when we consider "non-standard" interpretation structures. The use of $\text{INT}(\mathcal{S})$ instead of $\text{INT}_{st}(\mathcal{S})$ is justified by the following reason: by using $\text{INT}(\mathcal{S})$ we can take profit of the completeness theorem of CEQ, as will be done in Section 5.4 below.

5.2 Fibring non-truth functional logics

In this section we introduce fibring of interpretation systems presentations. Recall, from Chapter 3, the general idea when fibring two interpretation systems: their signatures are putting together, and the models are defined as being the structures over the new signature whose reducts to the given signatures are models of the respective systems.

Within the present framework, the models are two-sorted algebras (of formulas and truth values). It should be expected, therefore, that, in the fibring (as it was done in the truth functional case) the models would still be two-sorted algebras over the new signature whose reducts are models of the logics being fibred. When fibring two interpretation system presentations, thus, we expect to put together the signatures and the requirements on the valuation map.

Unconstrained fibring corresponds to the case where the signatures of the given systems are disjoint. Otherwise the fibring is constrained. Of course, a rigorous definition (in terms of category theory) of both form of fibring is possible, as we shall see.

The basic forms of fibring considered here lead to new logics that sometimes need small adjustments in order to cope with the application at hand. These adjustments consist of the adding of further interaction rules, written in the new signature.

Recall the category **Sig** presented in Remark 2.1.10. The category of interpretation systems is defined as follows:

Definition 5.2.1 An *interpretation system presentation* morphism

$$h : \langle C, S\rangle \to \langle C', S'\rangle$$

is a morphism $h : C \to C'$ in **Sig** such that, for each $s \in S$, $h(s)$ is in $S'^{\bullet}$.

Interpretation system presentations and its morphisms (with composition and identity maps inherited from **Sig**) constitute the category **Isp**. $\triangledown$

Remark 5.2.2 If $\mathcal{A}' = \langle \mathcal{A}'_\phi, \mathcal{A}'_\tau, \cdot_{\mathcal{A}'}\rangle$ is in $\text{INT}(\mathcal{S}')$ then its reduct to $\Sigma(C, \Xi)$ via h is the interpretation structure

$$\mathcal{A}'|^h_{\Sigma(C,\Xi)} = \langle (\mathcal{A}'_\phi)|^h_{\Sigma(C,\Xi)}, \mathcal{A}'_\tau, \cdot_{\mathcal{A}'|^h_{\Sigma(C,\Xi)}}\rangle$$

over $\Sigma(C,\Xi)$ such that, for every term t over $\Sigma(C,\Xi)$ and every assignment α over $\mathcal{A}'|^h_{\Sigma(C,\Xi)}$ (that is, every assignment α over $\mathcal{A}'$),

$$[\![t]\!]^\alpha_{\mathcal{A}'|^h_{\Sigma(C,\Xi)}} = [\![\hat{h}(t)]\!]^\alpha_{\mathcal{A}'}.$$

In particular, $[\![t]\!]_{\mathcal{A}'|^h_{\Sigma(C)}} = [\![\hat{h}(t)]\!]_{\mathcal{A}'}$ for every closed term t over $\Sigma(C)$.

Clearly, a morphism $h : \langle C, S\rangle \to \langle C', S'\rangle$ satisfies the following property: for every $\mathcal{A}' \in \textsc{Int}(\mathcal{S}')$, its reduct to $\Sigma(C)$ via h is in $\textsc{Int}(\mathcal{S})$. $\triangledown$

Definition 5.2.3 The *fibring of interpretation systems presentations* $\mathcal{S}' = \langle C', S'\rangle$ and $\mathcal{S}'' = \langle C'', S''\rangle$ is the interpretation system presentation

$$\mathcal{S}' \cup \mathcal{S}'' = \langle C, S\rangle$$

such that $C = C' \cup C''$ and $S = S' \cup S''$. $\triangledown$

As before, when $C' \cap C'' = \emptyset$ the fibring is said to be *unconstrained*, and it is said to be *constrained* otherwise.

The models of the fibred interpretation system presentations can be characterized as follows (details of the proof are left to the reader):

Proposition 5.2.4 *Given interpretation system presentations* $\mathcal{S}' = \langle C', S'\rangle$ *and* $\mathcal{S}'' = \langle C'', S''\rangle$, *an interpretation structure* $\mathcal{A}$ *over the signature* $\Sigma(C' \cup C'', \Xi)$ *belongs to* $\textsc{Int}(\mathcal{S}' \cup \mathcal{S}'')$ *if and only if:*

- $\mathcal{A}|_{\Sigma(C',\Xi)} \in \textsc{Int}(\mathcal{S}')$;
- $\mathcal{A}|_{\Sigma(C'',\Xi)} \in \textsc{Int}(\mathcal{S}'')$;

where $\mathcal{A}|_{\Sigma(C',\Xi)}$ *and* $\mathcal{A}|_{\Sigma(C'',\Xi)}$ *are the reducts of* $\mathcal{A}$ *to the signature* $\Sigma(C', \Xi)$ *and the signature* $\Sigma(C'', \Xi)$ *via the inclusions*

$$i' : \langle C', S'\rangle \to \langle C' \cup C'', \mathcal{S}' \cup \mathcal{S}''\rangle \text{ and } i'' : \langle C'', S''\rangle \to \langle C' \cup C'', \mathcal{S}' \cup \mathcal{S}''\rangle$$

respectively. $\triangledown$

From a categorial point of view, the unconstrained fibring is, as usual, a coproduct:

Proposition 5.2.5 *Let* $\mathcal{S}'$ *and* $\mathcal{S}''$ *be two interpretation system presentations. Then* $\mathcal{S}' \cup \mathcal{S}''$ *is the coproduct of* $\mathcal{S}$ *and* $\mathcal{S}''$ *in the category* ***Isp***.

Remark 5.2.6 With respect to the constrained fibring, the construction can be characterized as a cocartesian lifting by the forgetful functor $N : \mathbf{Isp} \to \mathbf{Sig}$ along the signature coequalizer for the pushout of the given signatures. We left the details of the construction to the interested reader. In Chapter 7 we will describe this construction. $\triangledown$

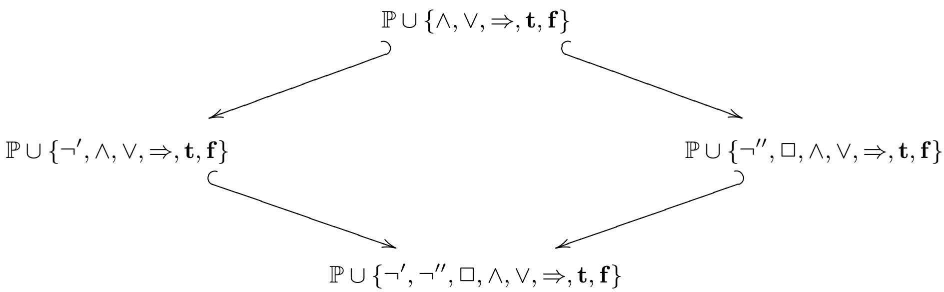


Figure 5.1: Fibring of signatures of $\mathfrak{C}_1$ and **KD**

The method for combining interpretation systems presentations just introduced is exemplified below.

Example 5.2.7 The paraconsistent deontic logic $\mathfrak{C}_1^D$ was introduced in [75]. This logic is an extension of the paraconsistent system $\mathfrak{C}_1$ by including a modal operator $\Box$ with the properties of the deontic ("obligatory") operator of the modal logic **KD**. It is natural to ask whether the logic $\mathfrak{C}_1^D$ is the fibring of $\mathfrak{C}_1$ and **KD**.

The obvious idea is to take the fibring of $\mathfrak{C}_1$ and **KD** constrained to the sharing of the propositional symbols in $\mathbb{P} \cup \{\wedge, \vee, \Rightarrow, \mathbf{t}, \mathbf{f}\}$.

Let $\mathcal{S}' = \langle C', S' \rangle$ be the interpretation system presentation for $\mathfrak{C}_1$ described in Example 5.1.9 but now defined in a signature containing the negation symbol $\neg'$ instead of $\neg$. On the other hand, let $\mathcal{S}'' = \langle C'', S'' \rangle$ be the interpretation system presentation for **KD** described in Example 5.1.14, but now including the negation symbol $\neg''$ instead of $\neg$. The resulting fibring by sharing $\mathbb{P} \cup \{\wedge, \vee, \Rightarrow, \mathbf{t}, \mathbf{f}\}$, as depicted in Figure 5.1, is

$$\mathcal{S}' \cup \mathcal{S}'' = \langle C' \cup C'', S' \cup S'' \rangle$$

where

- $(C' \cup C'')_0 = \mathbb{P} \cup \{\mathbf{t}, \mathbf{f}\}$;
- $(C' \cup C'')_1 = \{\neg', \neg'', \Box\}$;
- $(C' \cup C'')_2 = \{\wedge, \vee, \Rightarrow\}$;
- $(C' \cup C'')_k = \emptyset$ for $k > 2$.

It is worth noting that there are two negations in $C' \cup C''$: $\neg'$ coming from C' and $\neg''$ coming from C''. The former is the paraconsistent negation inherited from $\mathfrak{C}_1$ and the latter is the classical negation inherited from **KD**. Of course, in $C' \cup C''$ the expression γ° is an abbreviation of $(\neg'(\gamma \wedge (\neg' \gamma)))$.

It is easy to see that the fibring defined above is weaker than (the bivaluations presentation of) $\mathfrak{C}_1^D$. However, it is enough to add one additional meta-axiom on valuations to the fibred interpretation system presentation:

$$(\quad \longrightarrow (v(y_1^\circ) \leq v((\Box\, y_1)^\circ))).$$

Using the terminology introduced in Chapter 1, this procedure can be seen as a *splitting* of $\mathfrak{C}_1^D$ in the components **KD** and $\mathfrak{C}_1$. The idea of adding an extra interaction axiom is in the spirit of the original proposal of fibring, as introduced in [108]. $\triangledown$

There are some other interesting examples of combination of modal and paraconsistent reasoning in the literature, for instance [83, 84, 224]. In these references, using paraconsistent techniques, some problems of deontic logic having to do with deontic paradoxes and moral dilemmas are studied. All these systems could be, of course, analyzed from the point of view of fibring non-truth functional logics presented in this chapter.

We synthesize the properties of fibring of non-truth functional interpretation system presentations as follows:

- *homogeneous combination mechanism*: both original logics are presented by interpretation system presentations;

- *algorithmic combination of logics*: given the interpretation system presentations for the original logics, we know how to define the interpretation system presentation that corresponds to their fibring, but in many cases the given logics have to be pre-processed (that is, the interpretation system presentations have to be extracted);

- *canonical combination of logics*: the fibring is the minimal proof system among those that are stronger then the original proof systems.

5.3 Non-truth functional logic systems

In this section, as it was done in the previous chapters, an appropriate notion of logic system will be introduced. As done before, we first concentrate on the proof-theoretical counterpart of interpretation system presentations. Again, the proof-theory will be presented in terms of Hilbert calculi. We then consider logic systems which encompass both semantic and proof-theoretical aspects.

Recall, from Definition 2.3.1, the notion of Hilbert calculi which include global and local rules.

Continuing with the example of the deontic paraconsistent logic $\mathfrak{C}_1^D$, we consider the Hilbert calculus corresponding to the paraconsistent system $\mathfrak{C}_1$ and the Hilbert calculus corresponding to modal system **KD**.

Example 5.3.1 The Hilbert calculus for $\mathfrak{C}_1$ presented in Example 2.2.9 can be reformulated in terms of local and global rules, by considering all the rules as being simultaneously local and global: $R_g = R_\ell$. ▽

Example 5.3.2 Adapting from [153, 178], the modal Hilbert calculus for **KD** consists of the following set R_ℓ of rules:

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, (((\neg\, \xi_1) \Rightarrow (\neg\, \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\, \xi_1) \Rightarrow (\Box\, \xi_2))) \rangle$;
- $\langle \emptyset, ((\Box\, \xi_1) \Rightarrow (\neg(\Box(\neg\, \xi_1)))) \rangle$;
- $\langle \emptyset, ((\xi_1 \vee \xi_2) \Leftrightarrow ((\neg\, \xi_1) \Rightarrow \xi_2)) \rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Leftrightarrow (\neg((\neg\, \xi_1) \vee (\neg\, \xi_2)))) \rangle$;
- $\langle \emptyset, (\mathbf{t} \Leftrightarrow (\xi_1 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, (\mathbf{f} \Leftrightarrow (\xi_1 \wedge (\neg\, \xi_1))) \rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$.

On the other hand, $R_g = R_\ell \cup \{\langle \{\xi_1\}, (\Box\, \xi_1) \rangle\}$. ▽

Example 5.3.3 We now revisit modal paraconsistent logic. We can perform the fibring (as Hilbert calculi) of $\mathfrak{C}_1$ and **KD**, while sharing the propositional symbols in $\mathbb{P} \cup \{\wedge, \vee, \Rightarrow, \mathbf{t}, \mathbf{f}\}$. The resulting Hilbert calculus has all the global and local rules for both $\mathfrak{C}_1$ and **KD**, but defined in the mixed language. Clearly, the resulting calculus is weaker than the deontic paraconsistent system $\mathfrak{C}_1^D$ of [75] (at the proof-theoretic level). The calculus $\mathfrak{C}_1^D$ can be easily recovered, simply by adding to the fibred Hilbert calculus the following proof rule:

- $\langle \emptyset, (\xi_1^\circ \Rightarrow (\Box\, \xi_1)^\circ) \rangle$.

This interaction axiom, which is already present in $\mathfrak{C}_1^D$, could never be obtained using the basic fibring operation since it makes full use of the mixed language. This kind of interaction rules can never be obtained by using the current notion of (propositional) fibring. It is worth noting that the semantic counterpart of the interaction axiom added to the fibred Hilbert calculus was also added to the corresponding fibred interpretation system presentation in Example 5.2.7. ▽

As before, logic systems can be considered. Logic systems are structures which put together both semantic and proof-theoretic aspects. In the present framework, a logic system includes an interpretation system presentation and a Hilbert calculus. As mentioned in Chapter 3, logic systems provide the right setting to define metaproperties such as soundness and completeness.

Definition 5.3.4 A *logic system presentation* is a tuple

$$\mathcal{L} = \langle C, S, R_g, R_\ell \rangle$$

such that the pair $\langle C, S \rangle$ is an interpretation system presentation and the triple $\langle C, R_g, R_\ell \rangle$ is a Hilbert calculus. ▽

Again we can use $\Gamma \vdash^g_{\mathcal{L}} \varphi$ whenever $\Gamma \vdash^g_H \varphi$, where H is the underlying Hilbert calculus, and $\Gamma \vDash^g_{\mathcal{L}} \varphi$ whenever $\Gamma \vDash^g_{\mathcal{S}} \varphi$, where $\mathcal{S}$ is the underlying interpretation system presentation. The same notation applies to local reasoning.

As expected, the (unconstrained and constrained) fibring of logic systems is obtained by the corresponding fibring of the underlying interpretation system presentations and Hilbert calculi. The reader interested in category theory can easily check that, one more time, unconstrained fibrings of logic system presentations correspond to coproducts, and constrained fibrings of logic system presentations correspond to cocartesian liftings by the forgetful functor from the category of logic system presentations to **Sig**. These operations are performed in the category of logic system presentations, which is defined in the obvious way.

Example 5.3.5 The logic system presentations for $\mathfrak{C}_1$ and **KD** will be denoted by $\mathcal{L}_{\mathfrak{C}_1}$ and $\mathcal{L}_{\mathbf{KD}}$, respectively, and their fibring while sharing $\mathbb{P} \cup \{\wedge, \vee, \Rightarrow, \mathbf{t}, \mathbf{f}\}$ will be denoted by $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathbf{KD}}$ (recall Examples 5.2.7 and 5.3.3). ▽

Soundness and completeness of logic system presentations are defined as expected.

Definition 5.3.6 Let $\mathcal{L} = \langle C, S, R_g, R_\ell \rangle$ be a logic system presentation. We say that $\mathcal{L}$ is *sound* if, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$:

- $\Gamma \vdash^g_H \varphi$ implies $\Gamma \vDash^g_{\mathcal{S}} \varphi$;
- $\Gamma \vdash^\ell_H \varphi$ implies $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$.

On the other hand, $\mathcal{L}$ is *complete* if, for every $\Gamma \subseteq L(C)$ and $\varphi \in L(C)$:

- $\Gamma \vDash^g_{\mathcal{S}} \varphi$ implies $\Gamma \vdash^g_H \varphi$;
- $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$ implies $\Gamma \vdash^\ell_H \varphi$. ▽

Example 5.3.7 From the adequacy of the respective bivaluation semantics it is easy to prove that the logic system presentations $\mathcal{L}_{\mathfrak{C}_1}$ and $\mathcal{L}_{\mathbf{KD}}$ are sound and complete. ▽

5.4 Preservation results

In this section it is investigated the problem of preservation of soundness and completeness by fibring in the present framework. As it was done before, some sufficient conditions guaranteeing the preservation of soundness and completeness by fibring are established. At this point the use of the metalogic CEQ, as well as its soundness and completeness property (which can be easily obtained by adapting the proofs in [132, 208]) is crucial.

As we shall see in the proof of Proposition 5.4.11, the key idea is that the relevant part of the calculus of CEQ can be encoded in the object Hilbert calculus, provided that certain conditions are satisfied (recall Definitions 5.4.3 and 5.4.7).

5.4.1 Encoding CEQ in the object Hilbert calculus

A fundamental step towards the intended encoding is to investigate the inferential power of CEQ with respect to interpretation systems. This is done in Lemma 5.4.10. To begin with, recall the substitutions σ^+ and σ^- introduced in Definition 5.1.17.

Definition 5.4.1 Let $\mathcal{S} = \langle C, S \rangle$ be an interpretation system presentation. We define the following relations, where $\Gamma \cup \{\varphi\} \subseteq L(C)$:

- $\Gamma \vdash^g_{\mathcal{S}} \varphi$ if $S^\bullet \cup \{(\ \longrightarrow (v(\gamma) \approx \top)) : \gamma \in \Gamma\} \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\varphi) \approx \top)$;
- $\Gamma \vdash^\ell_{\mathcal{S}} \varphi$ if
$$S^\bullet \cup \{(\longrightarrow (v(\xi_1) \leq v(\sigma^+(\gamma)))) : \gamma \in \Gamma\} \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\xi_1) \leq v(\sigma^+(\varphi))). \qquad \nabla$$

Observe that the use of substitution σ^+ in the definition of $\vdash^\ell_{\mathcal{S}}$ above guarantees that the variable ξ_1 does not occur in the term $v(\sigma^+(\varphi))$.

The next result shows that the relations introduced above correspond, respectively, to global and local derivations in the sense of Definition 5.1.15

Proposition 5.4.2 *Given an interpretation system presentation $\mathcal{S} = \langle C, S \rangle$ and $\Gamma \cup \{\varphi\} \subseteq L(C)$, we have:*

- $\Gamma \vDash^g_{\mathcal{S}} \varphi$ *if and only if* $\Gamma \vdash^g_{\mathcal{S}} \varphi$;
- $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$ *if and only if* $\Gamma \vdash^\ell_{\mathcal{S}} \varphi$.

Proof. This is an immediate consequence of the completeness of CEQ. In the local case it is used, additionally, Proposition 5.1.18. $\triangleleft$

By considering the result above, it should be clear that soundness and completeness can now be expressed in terms of a relevant part of CEQ. This justifies the intended encoding of CEQ within the object Hilbert calculus.

The encoding of (a relevant part of) CEQ to be defined below requires the assumption of some additional properties of the logic system presentations. The technique is analogous to what was done in Chapters 2 and 3, by requiring the existence of some connectives with intuitionistic behavior.

Definition 5.4.3 A logic system presentation $\mathcal{L} = \langle C, S, R_g, R_\ell \rangle$ is said to be *rich* if:

- $\mathbf{t}, \mathbf{f} \in C_0$ and $\wedge, \vee, \Rightarrow \in C_2$;
- $S^\bullet \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\mathbf{t}) \approx \top)$;
- $S^\bullet \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\mathbf{f}) \approx \bot)$;
- $S^\bullet \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(y_1 \wedge y_2) \approx \sqcap(v(y_1), v(y_2)))$;
- $S^\bullet \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(y_1 \vee y_2) \approx \sqcup(v(y_1), v(y_2)))$;
- $S^\bullet \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(y_1 \Rightarrow y_2) \approx \sqsupset(v(y_1), v(y_2)))$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle \in R_\ell$. ∇

Example 5.4.4 The logic system presentations $\mathcal{L}_{\mathfrak{C}_1}$ and $\mathcal{L}_{\mathbf{KD}}$ (see Example 5.3.5), as well as many other logics occurring commonly in the literature, are rich. ∇

A rich logic system presentation allows to translate faithfully from the metalogic level to the object logic level in a natural way. In fact, a closed term of sort τ over $\Sigma(C, \Xi)$ is mapped to a formula in $L(C)$ by means of a mapping $*$ obeying the following rules:

$v(\gamma)^*$ is γ;

$\top^*$ is $\mathbf{t}$;

$\bot^*$ is $\mathbf{f}$;

$-(t)^*$ is $(t^* \Rightarrow \mathbf{f})$;

$\sqcap(t_1, t_2)^*$ is $(t_1^* \wedge t_2^*)$;

$\sqcup(t_1, t_2)^*$ is $(t_1^* \vee t_2^*)$;

$\sqsupset (t_1, t_2)^*$ is $(t_1^* \Rightarrow t_2^*)$.

Using the mapping $*$ defined above, a closed τ-equation $(t_1 \approx t_2)$ is naturally translated to the formula $(t_1 \approx t_2)^*$ given by $(t_1^* \Leftrightarrow t_2^*)$. Finally, if E is a set of closed τ-equations, then E^* will denote the set of formulas $\{\text{eq}^* : \text{eq} \in E\}$.

The main property of the translation $*$ is the following: if t is a truth value closed term then t^* is a formula having the truth value t through the valuation v (see Lemma 5.4.5). The proof of this fact is immediate from the definition of the mapping $*$ and the completeness of CEQ, assuming richness of the logic system presentation.

Lemma 5.4.5 *Let $\mathcal{L}$ be a rich logic system presentation and t a closed τ-term over $\Sigma(C, \Xi)$. Then:*

$$S^\bullet \vdash^{CEQ}_{\Sigma(C,\Xi)} (v(t^*) \approx t).$$

As a consequence of this, it is obtained an useful technical result.

Lemma 5.4.6 *Let $\mathcal{L}$ be a rich logic system presentation, t_1 and t_2 closed τ-terms over $\Sigma(C, \Xi)$ and $\mathcal{A} \in \textsc{Int}(\mathcal{S})$. Then:*

$$[\![t_1]\!]_\mathcal{A} \leq_\mathcal{A} [\![t_2]\!]_\mathcal{A} \quad \textit{if and only if} \quad v_\mathcal{A}([\![(t_1^* \Rightarrow t_2^*)]\!]_\mathcal{A}) = \top_\mathcal{A}$$

and

$$[\![t_1]\!]_\mathcal{A} = [\![t_2]\!]_\mathcal{A} \quad \textit{if and only if} \quad v_\mathcal{A}([\![(t_1^* \Leftrightarrow t_2^*)]\!]_\mathcal{A}) = \top_\mathcal{A}.$$

Proof. This is a direct consequence of Proposition 5.1.8, the previous lemma and the completeness of CEQ. ◁

In order to obtain the intended encoding of the relevant part of the metareasoning into the object calculus, an additional requirement (besides richness) is necessary:

Definition 5.4.7 A rich logic system presentation $\mathcal{L}$ is said to be *equationally appropriate* if

$$\{(\rho(\text{eq}_1))^*, \ldots, (\rho(\text{eq}_n))^*\} \vdash^g_\mathcal{L} (\rho(\text{eq}))^*$$

for every conditional equation $(\text{eq}_1 \ \& \ \ldots \ \& \ \text{eq}_n \longrightarrow \text{eq})$ in $S^\bullet$ and every closed substitution ρ. ▽

We are now ready to state the main results of this section, namely the relationship between completeness and equational appropriateness (recall Propositions 5.4.8 and 5.4.11). Since it is much easier to analyze the preservation by fibring of equational appropriateness than the preservation by fibring of completeness, the characterization of completeness in terms of equational appropriateness is very useful.

Proposition 5.4.8 *Every rich and complete logic system presentation is equationally appropriate.*

Proof. Let $\mathcal{L}$ be a rich and complete logic system presentation. Consider the structure $\mathcal{A} \in \text{INT}(\mathcal{S})$, and let $((t_1 \approx s_1) \& \ldots \& (t_n \approx s_n) \longrightarrow (t \approx s)$ be a conditional equation in $S^\bullet$ and ρ a closed substitution.

Suppose that $v_\mathcal{A}([\![(\rho(t_i))^* \Leftrightarrow (\rho(s_i))^*]\!]_\mathcal{A}) = \top_\mathcal{A}$ for $i = 1, \ldots, n$. Then, using Lemma 5.4.6,

$$[\![\rho(t_i)]\!]_\mathcal{A} = [\![\rho(s_i)]\!]_\mathcal{A}$$

for $i = 1, \ldots, n$. Consider the assignment given by $\alpha = [\![\_]\!]_\mathcal{A} \circ \rho$. It is easy to prove that $[\![\rho(r)]\!]_\mathcal{A} = [\![r]\!]^\alpha_\mathcal{A}$, for every τ-term r over $\Sigma(C, \Xi)$. In particular, $[\![t_i]\!]^\alpha_\mathcal{A} = [\![s_i]\!]^\alpha_\mathcal{A}$ for $i = 1, \ldots, n$. By definition of $\text{INT}(\mathcal{S})$, the interpretation structure $\mathcal{A}$ is a model of the given conditional equation of $S^\bullet$. Thus, it follows that $[\![t]\!]^\alpha_\mathcal{A} = [\![s]\!]^\alpha_\mathcal{A}$, that is,

$$[\![\rho(t)]\!]_\mathcal{A} = [\![\rho(s)]\!]_\mathcal{A}.$$

Using again Lemma 5.4.6, $v_\mathcal{A}([\![(\rho(t))^* \Leftrightarrow (\rho(s))^*]\!]_\mathcal{A}) = \top_\mathcal{A}$. This means that

$$\{((\rho(t_1))^* \Leftrightarrow (\rho(s_1))^*), \ldots, ((\rho(t_n))^* \Leftrightarrow (\rho(s_n))^*)\} \vDash^g_\mathcal{S} ((\rho(t))^* \Leftrightarrow (\rho(s))^*).$$

Using completeness, we get

$$\{((\rho(t_1))^* \Leftrightarrow (\rho(s_1))^*), \ldots, ((\rho(t_n))^* \Leftrightarrow (\rho(s_n))^*)\} \vdash^g_\mathcal{L} ((\rho(t))^* \Leftrightarrow ((\rho(s))^*).$$

In other words, the given logic system presentation is equationally appropriate. ◁

The converse of this theorem is proved in Proposition 5.4.11 below. Previous to this, it is necessary to establish some technical lemmas.

Lemma 5.4.9 *Let $\mathcal{L}$ be an equationally appropriate logic system presentation and $\Gamma \cup \{\varphi\}$ be a set of schema formulas where ξ_1 does not occur. If*

$$\{(\xi_1 \Rightarrow \gamma) : \gamma \in \Gamma\} \vdash^g_\mathcal{L} (\xi_1 \Rightarrow \varphi)$$

then $\Gamma \vdash^\ell_\mathcal{L} \varphi$.

Proof. The crucial point of the proof is that, since $S^\bullet$ must contain a specification of the class of the Heyting algebras, then every intuitionistic theorem written in the signature $\mathbf{t}, \mathbf{f}, \wedge, \vee, \Rightarrow$ must be provable in the Hilbert calculus of $\mathcal{L}$, because the equational appropriateness of $\mathcal{L}$. This fact will be used along the proof.

Thus, suppose that $\{(\xi_1 \Rightarrow \gamma) : \gamma \in \Gamma\} \vdash^g_\mathcal{L} (\xi_1 \Rightarrow \varphi)$. It follows that

$$\{(\xi_1 \Rightarrow \gamma_1), \ldots, (\xi_1 \Rightarrow \gamma_n)\} \vdash^g_\mathcal{L} (\xi_1 \Rightarrow \varphi)$$

for some finite set $\{\gamma_1, \ldots, \gamma_n\} \subseteq \Gamma$, given the finite character of derivations. Let γ be the schema formula $\gamma_1 \wedge \ldots \wedge \gamma_n$ and consider the schema variable substitution σ such that $\sigma(\xi_1) = \gamma$ and $\sigma(\xi_i) = \xi_i$ for every $i \geq 2$. By the structurality of proofs, and the fact that the schema variable ξ_1 does not occur in $\Gamma \cup \{\varphi\}$, we easily obtain that

$$\{(\gamma \Rightarrow \gamma_1), \ldots, (\gamma \Rightarrow \gamma_n)\} \vdash^g_\mathcal{L} (\gamma \Rightarrow \varphi).$$

By using straightforward intuitionistic reasoning it follows that $\vdash^g_{\mathcal{L}} (\gamma \Rightarrow \gamma_i)$ for $i = 1, \ldots, n$. Thus $\vdash^g_{\mathcal{L}} (\gamma \Rightarrow \varphi)$ and then, using again intuitionistic reasoning, we obtain $\vdash^g_{\mathcal{L}} (\gamma_1 \Rightarrow (\ldots \Rightarrow (\gamma_n \Rightarrow \gamma) \ldots))$. Finally, by modus ponens it follows that $\Gamma \vdash^\ell_{\mathcal{L}} \varphi$. ◁

Finally, it is necessary to analyze what can be proved in CEQ about interpretation systems:

Lemma 5.4.10 *Let $\mathcal{L}$ be an equationally appropriate logic system presentation, E a set of closed τ-equations over $\Sigma(C, \Xi)$ and ρ a closed substitution. If*

$$S^\bullet \cup \{(\longrightarrow eq) \ : \ eq \in E\} \vdash^{CEQ}_{\Sigma(C,\Xi)} (t_1 \approx t_2)$$

then either t_1, t_2 are the same term of sort ϕ, or t_1, t_2 are of sort τ and

$$E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*).$$

Proof. Recall the rules of CEQ in Definition 5.1.3.

Let ρ be a closed substitution. By induction on the length n of a proof in CEQ of $(t_1 \approx t_2)$ from $S^\bullet \cup \{(\longrightarrow \text{eq}) : \text{eq} \in E\}$ we will prove the intended result.

Base: $n = 1$.

(i) t_1 is $\rho'(s_1)$, t_2 is $\rho'(s_2)$ and $(t_1 \approx t_2)$ is obtained by modus ponens from the conditional equation $(\longrightarrow (s_1 \approx s_2)) \in S^\bullet$.
Then t_1 and t_2 are τ-terms. Moreover, it follows that

$$\vdash^g_{\mathcal{L}} ((\rho(\rho'(s_1)))^* \Leftrightarrow (\rho(\rho'(s_2)))^*)$$

by equational appropriateness. Then $E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$, by the monotonicity of provability.

(ii) t_1 is $\rho'(s_1)$, t_2 is $\rho'(s_2)$ and $(t_1 \approx t_2)$ is obtained by modus ponens from the conditional equation $(\longrightarrow (s_1 \approx s_2))$ with $(s_1 \approx s_2) \in E$.
Since both s_1 and s_2 are closed terms of sort τ then $\rho'(s_1)$ is s_1 and $\rho'(s_2)$ is s_2, and so t_1 and t_2 are closed terms of sort τ. Therefore, $(\rho(t_1))^* \Leftrightarrow (\rho(t_2))^* \in E^*$, that is, $(t_1^* \Leftrightarrow t_2^*) \in E^*$ and then

$$E^* \vdash^g_{\mathcal{L}} (t_1^* \Leftrightarrow t_2^*)$$

by the extensiveness of provability.

(iii) t_1 and t_2 are the same term, of either sort ϕ or τ, and $(t_1 \approx t_2)$ is obtained by reflexivity.
If the sort of the equation is ϕ, we are done. Otherwise, $(\rho(t_1))^*$ and $(\rho(t_2))^*$ are the same formula. By straightforward intuitionistic reasoning we conclude that $\vdash^g_{\mathcal{L}} (\xi_1 \Leftrightarrow \xi_1)$. Therefore,

$$E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$$

by the structurality and monotonicity of provability.

Step: $n > 1$.

(i) $(t_1 \approx t_2)$ is obtained from

$$S^\bullet \cup \{(\longrightarrow \text{eq}) : \text{eq} \in E\} \vdash^{\text{CEQ}}_{\Sigma(C,\Xi)} (t_2 \approx t_1)$$

by symmetry.
If the terms t_1 and t_2 have sort ϕ then, by induction hypothesis, they coincide. Otherwise, using again the induction hypothesis, we get $E^* \vdash^g_{\mathcal{L}} ((\rho(t_2))^* \Leftrightarrow (\rho(t_1))^*)$. By intuitionistic reasoning we know that $\vdash^g_{\mathcal{L}} ((\xi_1 \Leftrightarrow \xi_2) \Rightarrow (\xi_2 \Leftrightarrow \xi_1))$ and then $E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$.

(ii) $(t_1 \approx t_2)$ is obtained from

$$S^\bullet \cup \{(\longrightarrow \text{eq}) : \text{eq} \in E\} \vdash^{\text{CEQ}}_{\Sigma(C,\Xi)} \{(t_1 \approx t_3), (t_3 \approx t_2)\}$$

by transitivity.
If the terms t_1, t_2 and t_3 have sort ϕ then, by induction hypothesis, the three terms coincide. Otherwise, using again the induction hypothesis,

$$E^* \vdash^g_{\mathcal{L}} \{((\rho(t_1))^* \Leftrightarrow (\rho(t_3))^*), ((\rho(t_3))^* \Leftrightarrow (\rho(t_2))^*)\}.$$

On the other hand, straightforward intuitionistic reasoning give us

$$\{(\xi_1 \Leftrightarrow \xi_3), (\xi_3 \Leftrightarrow \xi_2)\} \vdash^g_{\mathcal{L}} (\xi_1 \Leftrightarrow \xi_2).$$

Therefore, $E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$.

(iii) t_1 is $f(t_{11}, \ldots, t_{1k})$, t_2 is $f(t_{21}, \ldots, t_{2k})$ and $(t_1 \approx t_2)$ is obtained from

$$S^\bullet \cup \{(\longrightarrow \text{eq}) : \text{eq} \in E\} \vdash^{\text{CEQ}}_{\Sigma(C,\Xi)} \{(t_{11} \approx t_{21}), \ldots, (t_{1k} \approx t_{2k})\}$$

by the rule of congruence.
If t_1 and t_2 have sort ϕ then $f \in C_k$ and so all the terms t_{ij} are of sort ϕ, by definition of $\Sigma(C, \Xi)$. Using the induction hypothesis, t_{1j} and t_{2j} must coincide (for $j = 1, \ldots, k$) and then t_1 coincide with t_2. Otherwise, f can either be v or a generator among $-, \sqcap, \sqcup, \sqsupset$, by definition of $\Sigma(C, \Xi)$. In the first case $k = 1$ and both t_{11} and t_{21} have sort ϕ. Then, by induction hypothesis, t_{11} coincides with t_{21} and so t_1 coincides with t_2. We can repeat step (iii) of the base to obtain

$$E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*).$$

Finally, if f is a truth value generator then all the terms t_{ij} are also of sort τ. Using the induction hypothesis,

$$E^* \vdash^g_{\mathcal{L}} \{((\rho(t_{11}))^* \Leftrightarrow (\rho(t_{21}))^*), \ldots, ((\rho(t_{1k}))^* \Leftrightarrow (\rho(t_{2k}))^*)\}.$$

On the other hand, by intuitionistic reasoning we have that

$$\{(\xi_1 \Leftrightarrow \xi_2), (\xi_3 \Leftrightarrow \xi_4)\} \vdash^g_{\mathcal{L}} ((\xi_1 \# \xi_3) \Leftrightarrow (\xi_2 \# \xi_4))$$

for $\# \in \{\wedge, \vee, \Rightarrow\}$. Then $E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$, since the generator $-$ is translated using $\mathbf{f}$ and $\Rightarrow$.

(iv) t_1 is $\rho'(s_1)$, t_2 is $\rho'(s_2)$ and $(t_1 \approx t_2)$ is obtained using the conditional equation $((s_{11} \approx s_{21})\ \&\ \ldots\ \&\ (s_{1k} \approx s_{2k}) \longrightarrow (s_1 \approx s_2)) \in S^\bullet$ from

$$S^\bullet \cup \{(\ \longrightarrow \mathrm{eq}\,) : \mathrm{eq} \in E\} \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} \{(\rho'(s_{11}) \approx \rho'(s_{21})), \ldots, (\rho'(s_{1k}) \approx \rho'(s_{2k}))\}$$

by modus ponens. In this case, all the terms s_{ij} have sort τ. Then, by induction hypothesis it follows that

$$E^* \vdash^g_{\mathcal{L}} \{((\rho(\rho'(s_{11})))^* \Leftrightarrow (\rho(\rho'(s_{21})))^*), \ldots, ((\rho(\rho'(s_{1k})))^* \Leftrightarrow (\rho(\rho'(s_{2k})))^*)\}.$$

On the other hand, by equational appropriateness we get

$$\Gamma \vdash^g_{\mathcal{L}} ((\rho(\rho'(s_1)))^* \Leftrightarrow (\rho(\rho'(s_2)))^*)$$

where $\Gamma = \{((\rho(\rho'(s_{11})))^* \Leftrightarrow (\rho(\rho'(s_{21})))^*), \ldots, ((\rho(\rho'(s_{1k})))^* \Leftrightarrow (\rho(\rho'(s_{2k})))^*)\}$. Therefore, $E^* \vdash^g_{\mathcal{L}} ((\rho(t_1))^* \Leftrightarrow (\rho(t_2))^*)$. ◁

Now we are ready to prove the converse of Proposition 5.4.8.

Proposition 5.4.11 *Every equationally appropriate logic system presentation is complete.*

Proof. Let $\mathcal{L}$ be an equationally appropriate logic system presentation and let $\Gamma \cup \{\varphi\} \subseteq L(C)$.

If $\Gamma \vDash^g_{\mathcal{S}} \varphi$ then, by Proposition 5.4.2, $\Gamma \vdash^g_{\mathcal{S}} \varphi$. That is,

$$S^\bullet \cup \{(\ \longrightarrow (v(\gamma) \approx \top)) : \gamma \in \Gamma\} \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\varphi) \approx \top).$$

Therefore, using Lemma 5.4.10, we have that

$$\{(\gamma \Leftrightarrow \mathbf{t}) : \gamma \in \Gamma\} \vdash^g_{\mathcal{L}} (\varphi \Leftrightarrow \mathbf{t}).$$

By intuitionistic reasoning we have that $\vdash^g_{\mathcal{L}} (\xi_1 \Leftrightarrow (\xi_1 \Leftrightarrow \mathbf{t}))$. Thus, it follows that $\Gamma \vdash^g_{\mathcal{L}} \varphi$.

On the other hand, if $\Gamma \vDash^\ell_{\mathcal{S}} \varphi$ then, using again Proposition 5.4.2, $\Gamma \vdash^\ell_{\mathcal{S}} \varphi$. That is,

$$S^\bullet \cup \{(\ \longrightarrow (v(\xi_1) \leq v(\sigma^+(\gamma)))) : \gamma \in \Gamma\} \vdash^{\mathrm{CEQ}}_{\Sigma(C,\Xi)} (v(\xi_1) \leq v(\sigma^+(\varphi))).$$

By Lemma 5.4.10,

$$\{((\xi_1 \wedge \sigma^+(\gamma)) \Leftrightarrow \xi_1) : \gamma \in \Gamma\} \vdash^g_{\mathcal{L}} ((\xi_1 \wedge \sigma^+(\varphi)) \Leftrightarrow \xi_1).$$

Using intuitionistic reasoning it can be proved that

$$\vdash^g_{\mathcal{L}} (((\xi_1 \wedge \xi_2) \Leftrightarrow \xi_1) \Leftrightarrow (\xi_1 \Rightarrow \xi_2))$$

and then it follows that

$$\{(\xi_1 \Rightarrow \sigma^+(\gamma)) : \gamma \in \Gamma\} \vdash^g_{\mathcal{L}} (\xi_1 \Rightarrow \sigma^+(\varphi)).$$

By Lemma 5.4.9 we infer that $\sigma^+(\Gamma) \vdash^\ell_{\mathcal{L}} \sigma^+(\varphi)$. Finally, using σ^- and structurality, it follows that $\Gamma \vdash^\ell_{\mathcal{L}} \varphi$. ◁

From Propositions 5.4.8 and 5.4.11, the equivalence between completeness and equational appropriateness for rich systems is obtained. This result will be used below for showing the preservation of completeness by fibring rich systems. It is worth noting that this equivalence may also be useful for establishing the completeness of logics endowed with a semantics presented by conditional equations. In fact, as mentioned above, it is a simpler task to verify equational appropriateness than to establish completeness.

5.4.2 Preservation of completeness by fibring

In this section we concentrate our attention on the preservation of soundness and of completeness by fibring.

As usual, the preservation of soundness is easier to state than the preservation of completeness:

Theorem 5.4.12 *Soundness is preserved by fibring.*

Proof. Let $\mathcal{L}$ be the fibring of two sound logic system presentations $\mathcal{L}'$ and $\mathcal{L}''$. Using Proposition 5.4.2, it is enough to prove the following: $\Gamma \vdash^g_{\mathcal{L}} \varphi$ implies that $\Gamma \vdash^g_{\mathcal{S}} \varphi$, and $\Gamma \vdash^\ell_{\mathcal{L}} \varphi$ implies that $\Gamma \vdash^\ell_{\mathcal{S}} \varphi$, for every $\Gamma \cup \{\varphi\} \subseteq L(C)$. Clearly, it is enough to prove the following that:

$$\mathit{Prem}(r) \vdash^g_{\mathcal{S}} \mathit{Conc}(r)$$

for every $r \in R_g$, and

$$\mathit{Prem}(r) \vdash^\ell_{\mathcal{S}} \mathit{Conc}(r)$$

for every $r \in R_\ell$. Thus, let $r \in R_g$. Assume, without loss of generality, that r is a global rule of $\mathcal{L}'$. Then $\mathit{Prem}(r) \vdash^g_{\mathcal{L}'} \mathit{Conc}(r)$, by definition of proof. Thus $\mathit{Prem}(r) \vdash^g_{\mathcal{S}'} \mathit{Conc}(r)$, by the soundness of $\mathcal{L}'$. That is,

$$S'^\bullet \cup \{(\longrightarrow (v(\gamma') \approx \top)) : \gamma' \in \mathit{Prem}(r)\} \vdash^{CEQ}_{\Sigma(C',\Xi)} (v(\mathit{Conc}(r)) \approx \top).$$

From this it follows that

$$S^\bullet \cup \{(\longrightarrow (v(\gamma') \approx \top)) : \gamma' \in \mathit{Prem}(r)\} \vdash^{CEQ}_{\Sigma(C,\Xi)} (v(\mathit{Conc}(r)) \approx \top),$$

that is, $\mathit{Prem}(r) \vdash^g_{\mathcal{S}} \mathit{Conc}(r)$. The proof for local derivations is similar. ◁

Finally, we address the problem of preservation of completeness by fibring. At this point the technical results concerning the encoding of the metalogic in the object Hilbert calculus will be used.

In order to obtain sufficient conditions for the preservation of completeness by fibring we concentrate the attention on rich systems. The first step is to prove the preservation of richness by fibring.

Lemma 5.4.13 *Richness is preserved by fibring provided that conjunction, disjunction, implication, true and false are shared.*

Proof. Since we are sharing conjunction, disjunction, implication, true and false, the conditions for richness concerning the signature and the valuation symbol are preserved. On the other hand, modus ponens is clearly a derivation rule in the fibring. ◁

The second requirement in order to guarantee the preservation of completeness by fibring is equational appropriateness. This property is also preserved by fibring, as we prove below:

Lemma 5.4.14 *Equational appropriateness is preserved by fibring provided that conjunction, disjunction, implication, true and false are shared.*

Proof. Let $\mathcal{L}'$ and $\mathcal{L}''$ be equationally appropriate logic system presentations, and let $\mathcal{L}$ be their fibring while sharing conjunction, disjunction, implication, true and false. By Definition 5.4.7 we know that $\mathcal{L}'$ and $\mathcal{L}''$ are rich and then, by Lemma 5.4.14, it follows that $\mathcal{L}$ is also rich.

Let

$$((t_1 \approx s_1) \; \& \; \ldots \; \& \; (t_n \approx s_n) \longrightarrow (t \approx s))$$

be a conditional equation in $S^\bullet$, and let ρ be a closed substitution. Clearly, by definition of fibring, this conditional equation belongs to some of the components, $S'^\bullet$ or $S''^\bullet$. Let us assume, without loss of generality, that the given conditional equation comes from $\mathcal{L}'$. Since, by hypothesis, $\mathcal{L}'$ is equationally appropriate, it follows that

$$\{((\rho'(t_1))^* \Leftrightarrow (\rho'(s_1))^*), \ldots, ((\rho'(t_n))^* \Leftrightarrow (\rho'(s_n))^*)\} \vdash^g_{\mathcal{L}'} ((\rho'(t))^* \Leftrightarrow (\rho'(s))^*),$$

where ρ' is the following closed substitution:

- for every $i \geq 1$, $\rho'(x_i) = v(\xi_{2i-1})$ and $\rho'(y_i) = \xi_{2i}$.

By definition of fibring of Hilbert calculi, it follows that

$$\{((\rho'(t_1))^* \Leftrightarrow (\rho'(s_1))^*), \ldots, ((\rho'(t_n))^* \Leftrightarrow (\rho'(s_n))^*)\} \vdash^g_{\mathcal{L}} ((\rho'(t))^* \Leftrightarrow (\rho'(s))^*).$$

Consider now the substitution σ on schema variables defined by:

- $\sigma(\xi_{2i-1}) = (\rho(x_i))^*$;

- $\sigma(\xi_{2i}) = \rho(y_i)$.

Using σ and the structurality of the Hilbert calculus of $\mathcal{L}$, it follows that

$$\sigma(\{((\rho'(t_1))^* \Leftrightarrow (\rho'(s_1))^*), \ldots, ((\rho'(t_n))^* \Leftrightarrow (\rho'(s_n))^*)\}) \vdash^g_{\mathcal{L}} \sigma((\rho'(t))^* \Leftrightarrow (\rho'(s))^*).$$

Now, by induction on complexity it is easy to prove that

$$\sigma((\rho'(u'))^*) = (\rho(u'))^*$$

for every term u' of sort τ over $\Sigma(C', \Xi)$. Thus

$$\{((\rho(t_1))^* \Leftrightarrow (\rho(s_1))^*), \ldots, ((\rho(t_n))^* \Leftrightarrow (\rho(s_n))^*)\} \vdash^g_{\mathcal{L}} ((\rho(t))^* \Leftrightarrow (\rho(s))^*).$$

From this, it follows that $\mathcal{L}$ is equationally appropriate. ◁

Theorem 5.4.15 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be rich, sound and complete logic system presentations. Then, their fibring $\mathcal{L}$ while sharing conjunction, disjunction, implication, true and false is also a sound and complete logic system presentation.*

Proof. The preservation of soundness follows from Proposition 5.4.12. On the other hand, the preservation of completeness is an immediate consequence of Lemma 5.4.14 and the equivalence between equational appropriateness and completeness for rich systems stated above. In more detail, suppose that $\mathcal{L}'$ and $\mathcal{L}''$ are two rich and complete logic system presentations, and let $\mathcal{L}$ be their fibring while sharing conjunction, disjunction, implication, true and false. Using Proposition 5.4.8, the systems $\mathcal{L}'$ and $\mathcal{L}''$ are also equationally appropriate. By Lemma 5.4.14, the logic system presentation $\mathcal{L}$ is equationally appropriate. Finally, by Proposition 5.4.11, the system $\mathcal{L}$ is complete. ◁

Example 5.4.16 Consider again the logic system presentations $\mathcal{L}_{\mathfrak{C}_1}$ and $\mathcal{L}_{\mathbf{KD}}$ (see Example 5.3.5). Since both are rich, sound and complete (see Examples 5.4.4 and 5.3.7) then their fibring $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathbf{KD}}$ while sharing $\mathbb{P} \cup \{\wedge, \vee, \Rightarrow, \mathbf{t}, \mathbf{f}\}$ (see Example 5.3.5) is also sound and complete, by Theorem 5.4.15. This system is a new modal paraconsistent logic system presentation, weaker than the paraconsistent deontic logic $\mathfrak{C}_1^D$ of [75], as observed above. It is worth noting that if we add to $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathbf{KD}}$:

- $(v(y_1^\circ) \leq v((\Box\, y_1)^\circ))$ as a valuation axiom; and
- $\langle \emptyset, (\xi_1^\circ \Rightarrow (\Box\, \xi_1)^\circ)\rangle$ as an axiom in the Hilbert calculus

(recalling that γ° is now an abbreviation for $(\neg'(\gamma \wedge (\neg'\, \gamma))))$ it is obtained a sound and complete logic system presentation that is equivalent to the system $\mathfrak{C}_1^D$, both at the proof-theoretic and the semantic levels. ▽

5.5 Self-fibring and non-truth functionality

This section analyzes self-fibring (that is, fibring two copies of the same logic) in the context of non-truth functionality.

The problem of obtaining the self-fibring of a given logic could appear, at first sight, as a trivial and uninteresting one: the first thing that comes to mind is that the self-fibring of a given logic produces the same logic, written in a signature with duplicate symbols, which collapse. Thus, for instance, the self-fibring of modal logic **KD** produces again **KD**, but with two symbols $\neg$ and $\neg'$ for negation, two symbols $\wedge$ and $\wedge'$ for conjunction, two symbols $\Box$ and $\Box'$ for necessitation, and so on (for simplicity, let us suppose that the symbols in C_0 are shared). The connectives should be equivalent in the resulting logic. Moreover, by the replacement property, any formula φ written with connectives $\wedge$, $\vee$, $\Rightarrow$, $\neg$ and $\Box$ should be equivalent to the formula φ' obtained from φ by replacing (some) occurrences of $\#$ by $\#'$, for $\# \in \{\wedge, \vee, \Rightarrow, \neg, \Box\}$.

This result is indeed true, as we shall see, as long as truth functional connectives are involved. However, if a logic contains a non-truth functional connective, say c, then the two versions of that connective in the self-fibring, c and c', are no longer necessarily equivalent. Moreover, if the logic does not satisfy the replacement property (as happens, for instance, with most of the **LFI**s, recall Example 5.1.10) then a formula φ could not be equivalent to the formula φ' obtained by replacing (some) occurrences of a connective c by its duplicate c', despite c being truth functional or not. In particular, this phenomenon occurs with the paraconsistent logic $\mathfrak{C}_1$.

Let us begin by analyzing the self-fibring of interpretation system presentations. Thus, let $\mathcal{S}$ be a truth functional interpretation system presentation, that is, one in which all the connectives are truth functional (possibly derived) connectives. In this special case, the self-fibring of $\mathcal{S}$ without sharing of connectives (just sharing the propositional symbols in $\mathbb{P}$) can be seen as the fibring $\mathcal{S} \cup \mathcal{S}'$ of $\mathcal{S}$ with a disjoint copy $\mathcal{S}'$ of $\mathcal{S}$. That is, $\mathcal{S}'$ is obtained from $\mathcal{S}$ by replacing each symbol $c \notin \mathbb{P}$ of C by c'. As mentioned above, $\mathcal{S} \cup \mathcal{S}'$ is a new version of $\mathcal{S}$ in which each connective appears duplicate. More precisely, if $c \in C_k$ (for $k > 0$) and c' is its duplicate then

$$v_{\mathcal{A}}([\![(c(t_1, \ldots, t_k))]\!]^{\alpha}_{\mathcal{A}}) = v_{\mathcal{A}}([\![(c'(t_1, \ldots, t_k))]\!]^{\alpha}_{\mathcal{A}})$$

for every interpretation structure $\mathcal{A}$, every assignment α over $\mathcal{A}$ and every terms $t_1, \ldots, t_k$ of sort ϕ. Additionally, if $c \in C_0$ is a constant symbol then

$$v_{\mathcal{A}}([\![c]\!]_{\mathcal{A}}) = v_{\mathcal{A}}([\![c']\!]_{\mathcal{A}}).$$

If we consider now a truth functional, rich and complete logic system presentation $\mathcal{L}$ then the self-fibring $\mathcal{L} \cup \mathcal{L}'$ of $\mathcal{L}$ with itself, while sharing C_0, conjunction, disjunction and implication, the situation is the same. That is, the formulas $(c(t_1, \ldots, t_k))$ and $(c'(t_1, \ldots, t_k))$ will be equivalent in $\mathcal{L} \cup \mathcal{L}'$, even if c and c' are not explicitly shared. Moreover, c can be replaced by c' in any formula, obtaining a formula equivalent to the original.

Consider now an interpretation system presentation $\mathcal{S}$ having a non-truth functional connective c. In this case, $v_{\mathcal{A}}([\![(c(t_1,\ldots,t_k))]\!]^{\alpha}_{\mathcal{A}})$ and $v_{\mathcal{A}}([\![(c'(t_1,\ldots,t_k))]\!]^{\alpha}_{\mathcal{A}})$ do not necessarily coincide for every interpretation structure $\mathcal{A}$ for $\mathcal{S}\cup\mathcal{S}'$. This means that, in general, $(c(t_1,\ldots,t_k))$ and $(c'(t_1,\ldots,t_k))$ are not equivalent in $\mathcal{L}\cup\mathcal{L}$.

Let us consider a concrete example. Let $\mathcal{S}$ be the interpretation system presentation given in Example 5.1.9, associated to (the semantic version of) the paraconsistent logic $\mathfrak{C}_1$. Consider the self-fibring $\mathcal{S}\cup\mathcal{S}'$ of $\mathcal{S}$ while sharing the symbols in C_0. Then, two families of connectives are obtained: $\{\wedge,\vee,\Rightarrow,\neg\}$ and $\{\wedge',\vee',\Rightarrow',\neg'\}$. Any model for $\mathcal{S}\cup\mathcal{S}'$ produces a valuation map $v_{\mathcal{A}}$ such that

$$v_{\mathcal{A}}([\![(t_1\sharp t_2)]\!]^{\alpha}_{\mathcal{A}}) = v_{\mathcal{A}}([\![(t_1\sharp' t_2)]\!]^{\alpha}_{\mathcal{A}})$$

for $\sharp\in\{\wedge,\vee,\Rightarrow\}$, since all the connectives above are truth functional.

In contrast, $v_{\mathcal{A}}([\![\neg t]\!]^{\alpha}_{\mathcal{A}})$ does not coincide necessarily with $v_{\mathcal{A}}([\![\neg' t]\!]^{\alpha}_{\mathcal{A}})$. For instance, consider two $\mathfrak{C}_1$-bivaluations v_1 and v_2 such that

- $v_1(p) = v_2(p)$ for every propositional symbol $p\in C_0$;
- $v_1(p_1) = v_1(\neg p_1) = 1$ and $v_2(p_1) = 1$, $v_2(\neg p_1) = 0$.

From this, it is possible to obtain a standard interpretation $\mathcal{A}$ presented by $\mathcal{S}\cup\mathcal{S}$ (recall Definition 5.1.16) such that

- $\mathcal{A}_\tau = \{0,1\}$;
- $\langle\mathcal{A}_\tau,\top_{\mathcal{A}},\bot_{\mathcal{A}},-_{\mathcal{A}},\sqcap_{\mathcal{A}},\sqcup_{\mathcal{A}},\sqsupset_{\mathcal{A}}\rangle$ coincides with $\mathit{2}$;
- $[\![\xi]\!]_{\mathcal{A}} = \xi$ for every $\xi\in\Xi$;
- $v_{\mathcal{A}} : L(C\cup C')\to\{0,1\}$ is defined recursively as follows:
  - $v_{\mathcal{A}}(p) = v_1(p) = v_2(p)$ for every propositional symbol $p\in C_0$;
  - $v_{\mathcal{A}}(\xi_k) = v_1(p_k) = v_2(p_k)$ for every $k\geq 1$;
  - $v_{\mathcal{A}}(\varphi\#\psi) = v_{\mathcal{A}}(\varphi)\#_{\mathcal{A}}v_{\mathcal{A}}(\psi)$ for every $\#\in\{\wedge,\wedge',\vee,\vee',\Rightarrow,\Rightarrow'\}$, where $\wedge_{\mathcal{A}} = \wedge'_{\mathcal{A}} = \sqcap_{\mathcal{A}}$, $\vee_{\mathcal{A}} = \vee'_{\mathcal{A}} = \sqcup_{\mathcal{A}}$ and $\Rightarrow_{\mathcal{A}} = \Rightarrow'_{\mathcal{A}} = \sqsupset_{\mathcal{A}}$;
  - $v_{\mathcal{A}}(\neg\varphi) = v_1(\neg[\![\varphi]\!]_{\mathcal{A}})$, for $\varphi\in L(C)$;
  - $v_{\mathcal{A}}(\neg\varphi) = 1$ if $v_{\mathcal{A}}(\varphi) = 0$, for $\varphi\in L(C\cup C')\setminus L(C)$;
  - $v_{\mathcal{A}}(\neg'\varphi) = v_2(\neg[\![\varphi]\!]_{\mathcal{A}})$, for $\varphi\in L(C')$;
  - $v_{\mathcal{A}}(\neg'\varphi) = 1$ if $v_{\mathcal{A}}(\varphi) = 0$, for $\varphi\in L(C\cup C')\setminus L(C')$.

It is easy to check that $\mathcal{A}$ is in fact a standard interpretation structure such that $v_{\mathcal{A}} : L(C\cup C')\to\{0,1\}$ is a mapping satisfying the following:

- $v_{\mathcal{A}}$ restricted to $L(C,\emptyset)$, the fragment generated by $\mathbb{P}\cup\{\mathbf{t},\mathbf{f},\wedge,\vee,\Rightarrow,\neg\}$, coincides with v_1;

- $v_{\mathcal{A}}$ restricted to $L(C', \emptyset)$, the fragment generated by $\mathbb{P} \cup \{\mathbf{t}, \mathbf{f}, \wedge', \vee', \Rightarrow', \neg'\}$, coincides with v_2.

On the other hand, $v_{\mathcal{A}}(\neg p_1) \neq v_{\mathcal{A}}(\neg' p_1)$, and then $\neg$ and $\neg'$ do not collapse in the resulting logic. By Theorem 5.4.15, $\neg\varphi$ and $\neg'\varphi$ are not equivalent formulas (unless they are both theorems).

The example above shows that the self-fibring $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathfrak{C}_1}$ of $\mathfrak{C}_1$ while sharing $\mathbb{P} \cup \{\mathbf{t}, \mathbf{f}\}$ produces two disjoint copies of $\mathfrak{C}_1$.

Despite the truth functional connectives $\wedge$, $\vee$ and $\Rightarrow$ collapse respectively with $\wedge'$, $\vee'$ and $\Rightarrow'$, they are not the same in the resulting logic $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathfrak{C}_1}$ as we now explain. If we replace some occurrences of $\#$ by $\#'$ in a formula, then the resulting formula could be not equivalent to the original one. More precisely: for every $\# \in \{\wedge, \vee, \Rightarrow\}$ there exists a schema formula φ such that $\sigma(\varphi)$ and $\sigma'(\varphi)$ are not equivalent in $\mathcal{L}_{\mathfrak{C}_1 \oplus \mathfrak{C}_1}$. Herein, σ and σ' are substitutions from Ξ to $L(C \cup C')$ such that

$$\sigma(\xi_1) = (\xi_1 \# \xi_2) \text{ and } \sigma'(\xi_1) = (\xi_1 \#' \xi_2).$$

Take, for instance, $\varphi = (\neg \xi_1)$, and consider a standard interpretation $\mathcal{A}$ as above, assuming that $v_1(p_1 \Rightarrow p_2) = v_1(\neg(p_1 \Rightarrow p_2)) = 1$. Note that $v_{\mathcal{A}}(\neg(\xi_1 \Rightarrow \xi_2)) = 1$ whereas $v_{\mathcal{A}}(\neg(\xi_1 \Rightarrow' \xi_2)) = 0$ since $v_{\mathcal{A}}(\xi_1 \Rightarrow' \xi_2) = v_2(p_1 \Rightarrow p_2) = 1$. Then,

$$\sigma(\varphi) = (\neg(\xi_1 \Rightarrow \xi_2)) \text{ and } \sigma'(\varphi) = (\neg(\xi_1 \Rightarrow' \xi_2))$$

are not equivalent.

The same result applies to the other truth functional connectives. In other words, the equivalence of $(\xi_1 \# \xi_2)$ and $(\xi_1 \#' \xi_2)$ does not guarantee the equivalence of $(\neg(\xi_1 \# \xi_2))$ and $(\neg(\xi_1 \#' \xi_2))$. This kind of phenomenon occurs in most of the **LFI**s studied in the literature, since commonly they do not satisfy the replacement property (see [49]). This shows that the self-fibring of logics without the replacement property may produce unexpected results.

5.6 Final remarks

This chapter addressed the problem of fibring non-truth functional logics. Such logics have the property that the denotation of formulas is not structural or, in more technical terms, that the denotation of formulas is not an homomorphism between algebras. In a truth functional logic we have that the denotation of formula $(c(\varphi_1, \ldots, \varphi_k))$ over model m is

$$[\![(c(\varphi_1, \ldots, \varphi_k))]\!]_m = \nu_k(c)([\![\varphi_1]\!]_m, \ldots, [\![\varphi_k]\!]_m).$$

Non-truth functional logics do not enjoy some nice properties like congruence and substitution of equivalents. This raises problems, as for instance in what concerns the use of standard proofs for completeness like the Lindenbaum-Tarski technique in the algebraic setting. Herein, we use a metaframework (conditional equational logic) for encoding such logics and take advantage of the completeness of this logic to prove completeness of non-truth functional logics.

The better known examples of non-truth functional logics are logics of formal inconsistency (**LFI**s) that also have the property that from a contradiction we do not necessarily infer everything (recall Example 5.1.10).

Another interesting application of the present setting, which deserves future research, is the possibility of applying the fibring of interpretation system presentations to a larger class of logics, by using Wójcicki's Reduction (see [278]) and Suszko's Reduction (see [194]). These techniques applied together imply that every standard logic admits a bivaluation semantics. Moreover, in [35] was presented a technique to obtain a bivaluation semantics for a large class of finite valued matrix logics. The obtained valuation semantics is defined by a set of clauses that can be easily transformed into interpretation system presentations. Thus, the techniques introduced in this chapter could be applied to combine (finite valued) matrix logics. An interesting question to be studied is the relationship between this method and the plain fibring of matrix logics described in Chapter 9.

Chapter 6

Fibring first-order logics

So far we have considered propositional based logics. However, sometimes, propositional based logics are not expressive enough for our purposes. We have then to work with more expressive logics, such as first-order or even higher order logics. Clearly, fibring mechanisms are still useful in these settings. We postpone the study of fibring higher-order logics to Chapter 7 and concentrate herein in the study of fibring first-order based logics.

In the first-order setting signatures may include connectives, as in propositional signatures, but they may also include function symbols, predicate symbols and quantifiers (variable binding operators). Two kinds of variables are now involved in the language: schema variables, as before, and quantification variables, that is, the variables that can be bind by the quantifiers.

From the deductive point of view, Hilbert calculi are adopted. The presence of quantifiers in first-order languages introduces some new problems. In particular, the substitution of schema variables within the scope of quantifiers may have unexpected and undesirable consequences. Hence, substitutions have to be carefully handled when using inference rules in a derivation. To deal with this problem we introduce the notion of proviso. Each inference rule includes a proviso whose purpose is to ensure a safe use of substitutions when the rule is applied.

With respect to semantics, things also become more elaborate in this first-order setting. Besides the denotation of connectives, we have to deal with the denotation of functions and predicate symbols, as well as the denotation of quantifiers. Therefore, we need semantic structures that, for instance, have to encompass both the semantics of quantifiers and the semantics of modal operators. We have in mind a powerset algebraic semantics recognizing that quantifiers can be seen as modalities. In this perspective, we adopt semantic structures endowed with a set of "points", a set of assignments to quantification variables and a set of "worlds", together with maps that associate to each "point" an assignment and a "world". As a result, quantifiers become special kind of modalities for which assignments play the role of worlds. Herein we take a different approach from the one proposed in [168] for studying fusion of first-order modal logics.

In Section 6.1 we introduce first-order based signatures and the corresponding languages. Next, in Section 6.2, we introduce interpretation structures and interpretation systems. First-order Hilbert calculi are presented in Section 6.3. Section 6.4 introduces first-order logic systems. In Section 6.5, we define fibring of first-order based logics. We illustrate the constructions with the case of classical first-order logic and modal classical first-order logic. The preservation by fibring of several methateorems as well as the preservation of completeness is discussed in Section 6.6. Finally, in Section 6.7, we make some final remarks.

This chapter capitalizes in [242] for the most part. It is also worthwhile to take a look at [241] in [112] as a preliminary investigation on this topic and also to understand better the problems raised by rules with provisos.

6.1 First-order signatures

This section introduces first-order based signatures and the corresponding first-order based languages.

The notion of signature considered in the previous chapters is not rich enough to cope with first-order features. Hence, we have to consider a more sophisticated notion of signature. Besides connectives, first-order based signatures include function and predicate symbols and variable binding operators, usually referred as quantifiers. For technical reasons to be detailed later on, we also include individual symbols as distinct from 0-ary function symbols (constants). Moreover, modalities are herein distinguished from the other connectives.

In what concerns the variables, we have to consider in this setting two kinds of variables. To begin with, we have to consider quantification variables, that is, the variables that quantifiers bind. Then, as before, we consider schema variables. Since first-order based languages include both terms and formulas, we distinguish between term schema variables and formula schema variables. Hence, we assume fixed throughout this chapter the following three denumerable pairwise disjoint sets of variables:

- $X = \{x_1, x_2, \ldots\}$;
- $\Theta = \{\theta_1, \theta_2, \ldots\}$;
- $\Xi = \{\xi_1, \xi_2, \ldots\}$.

The set X is the set of *quantification variables*, the set Θ is the set of *term schema variables* and the set Ξ is the set of *formula schema variables*.

We also assume as fixed the *equality symbol* $\approx$ and the *inequality symbol* $\not\approx$.

Next, we introduce the notion of first-order based signature where $\mathbb{N}^+$ is the set of all natural numbers greater than 0.

Definition 6.1.1 A *first-order signature* is a tuple

$$\Sigma = \langle Ind, F, P, C, Q, O \rangle$$

where

- Ind is a set;

and

- $F = \{F_k\}_{k \in \mathbb{N}}$;
- $P = \{P_k\}_{k \in \mathbb{N}}$;
- $C = \{C_k\}_{k \in \mathbb{N}}$;
- $Q = \{Q_k\}_{k \in \mathbb{N}^+}$;
- $O = \{O_k\}_{k \in \mathbb{N}^+}$;

are families of sets. $\triangledown$

The elements of Ind are the *individual symbols*. For each $k \in \mathbb{N}$, the elements of each F_k, P_k and C_k are respectively the *function symbols* of arity k, the *predicate symbols* of arity k and the *connectives* of arity k. For each $k \in \mathbb{N}^+$, the elements of each Q_k and O_k are respectively the *quantifiers* of arity k and the *modalities* of arity k. Observe that we do not include modalities in the family of connectives C, as we have done in the previous chapters, but, instead, we consider a distinct family O of modalities.

We assume that all the sets are pairwise disjoint and also disjoint from X, Θ and Ξ.

First-order signatures can be compared as expected: we say that Σ is *weaker than* Σ', written

$$\Sigma \leq \Sigma'$$

whenever

- $Ind \subseteq Ind'$;
- $F_k \subseteq F'_k$, $P_k \subseteq P'_k$ and $C_k \subseteq C'_k$, for every $k \in \mathbb{N}$;
- $Q_k \subseteq Q'_k$ and $O_k \subseteq O'_k$, for every $k \in \mathbb{N}^+$.

Example 6.1.2 An example of first-order signature, corresponding to a classical first-order signature, is $\Sigma = \langle Ind, F, P, C, Q, O \rangle$ where $Ind = \emptyset$, F and P are families of sets with $P_0 = \emptyset$, $O_k = \emptyset$ for $k \in \mathbb{N}^+$ and

- $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow\}$ and $C_k = \emptyset$ for $k = 0$ or $k > 2$;
- $Q_1 = \{(\forall x) : x \in X\}$ and $Q_k = \emptyset$ for $k > 1$.

The abbreviations for $\wedge$, $\vee$ and $\Leftrightarrow$ are the usual ones. We consider also the usual abbreviation for the existential quantifier $((\exists x)\varphi) =_{\text{def}} (\neg((\forall x)(\neg\,\varphi)))$. ▽

Example 6.1.3 Another example of first-order based signature, corresponding to a modal classical first-order signature, is $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O\rangle$ where *Ind* is a set, F and P are families of sets and

- $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow\}$ and $C_k = \emptyset$ for $k = 0$ or $k > 2$;
- $Q_1 = \{(\forall x) : x \in X\}$ and $Q_k = \emptyset$ for $k > 1$;
- $O_1 = \{\Box\}$ and $O_k = \emptyset$ for $k > 1$.

The abbreviation for $\Diamond$ is the usual one. ▽

Since in this chapter we only deal with first-order signatures, for simplicity, we will often use from now on"signature" instead of "first-order signature". In the remain of this section we consider the signature $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O\rangle$.

We now define the language over a given signature. First-order languages include both terms and formulas. We start by defining the terms.

Definition 6.1.4 The set $T(\Sigma)$ is inductively defined as follows:

- $\mathit{Ind} \cup F_0 \cup X \cup \Theta \cup \{\theta^{x'_1,\ldots,x'_k}_{\theta'_1,\ldots,\theta'_k} : \theta, \theta'_1, \ldots, \theta'_k \in \Theta,\ x'_1, \ldots, x'_k \in X\} \subseteq T(\Sigma)$;
- $f(t_1, \ldots, t_k) \in T(\Sigma)$ if $t_1, \ldots, t_k \in T(\Sigma)$ and $f \in F_k$, $k \in \mathbb{N}$. ▽

Each element of $T(\Sigma)$ is a *term*. A *ground term* is a term that does not involve schema variables. The set of all ground terms in $T(\Sigma)$ is denoted by $gT(\Sigma)$. Ground terms without quantification variables are said to be *closed terms*. The set of all closed terms in $T(\Sigma)$ is denoted by $cT(\Sigma)$.

Example 6.1.5 Recall the first-order signature Σ presented in Example 6.1.3. Assuming that f_1 and f_2 are functions symbols of arity 1 and 2 respectively,

$$x_1 \qquad f_1(x_1) \qquad {\theta_1}^{x_1}_{\theta_2} \qquad f_2(x_1, \theta_1)$$

are examples of terms in $T(\Sigma)$. The first two terms are ground terms. ▽

We now define the language of formulas.

Definition 6.1.6 The set $L(\Sigma)$ is inductively defined as follows:

- $P_0 \cup C_0 \cup \Xi \cup \{\xi^{x'_1,\ldots,x'_k}_{\theta'_1,\ldots,\theta'_k} : \xi \in \Xi,\ \theta'_1, \ldots, \theta'_k \in \Theta,\ x'_1, \ldots, x'_k \in X\} \subseteq L(\Sigma)$;
- $p(t_1, \ldots, t_k) \in L(\Sigma)$ if $t_1, \ldots, t_k \in T(\Sigma)$ and $p \in P_k$, $k \in \mathbb{N}^+$;

- $(t_1 \approx t_2) \in L(\Sigma)$ and $(t_1 \not\approx t_2) \in L(\Sigma)$ if $t_1, t_2 \in T(\Sigma)$;
- $(c(\varphi_1, \ldots, \varphi_k)) \in L(\Sigma)$ if $\varphi_1, \ldots, \varphi_k \in L(\Sigma)$ and $c \in C_k$, $k \in \mathbb{N}^+$;
- $(o(\varphi_1, \ldots, \varphi_k)) \in L(\Sigma)$ if $\varphi_1, \ldots, \varphi_k \in L(\Sigma)$ and $o \in O_k$, $k \in \mathbb{N}^+$;
- $((qx)(\varphi_1, \ldots, \varphi_k)) \in L(\Sigma)$ if $q \in Q_k$ and $x \in X$, $k \in \mathbb{N}^+$. $\triangledown$

Each element of $L(\Sigma)$ is a *formula*. As usual, a *ground formula* is a formula that does not involve schema variables. The set of all ground formulas in $L(\Sigma)$ is denoted by $gL(\Sigma)$. A formula is said to be *atomic* if does not involve connectives, quantifiers or modalities.

An occurrence of a quantification variable x in a formula is said to be a *free occurrence of* x in the formula if it is neither within the scope of a quantifier (qx) nor in the list $x'_1, \ldots, x'_k$ of the formula $\xi^{x'_1,\ldots,x'_k}_{\theta'_1,\ldots,\theta'_k}$.

In this first-order setting it is also relevant to consider *closed formulas*. A closed formula is a ground formula where there are no free occurrences of any quantification variable, that is, each occurrence of each $x \in X$ occurs within the scope of a quantifier (qx). The set of all closed formulas in $L(\Sigma)$ is denoted by $cL(\Sigma)$.

As before, we will often use infix notation when writing formulas involving connectives of arity 2. We can also write just qx instead of (qx) if no confusion arises.

Example 6.1.7 Recall the first-order signature Σ presented in Example 6.1.3. Assuming that p_1 is a predicate symbol of arity 1,

$$p_1(x_1) \quad \text{and} \quad ((\theta_1 \not\approx \theta_2) \Rightarrow (\neg(\theta_1 \approx \theta_2)))$$

are examples of formulas in $L(\Sigma)$. The first one is a ground atomic formula. The following

$$(\Box(\forall x_1\, p_1(x_1))) \quad \text{and} \quad ((\forall x_1\, \xi_1) \Rightarrow {\xi_1}^{x_1}_{\theta_1})$$

are also examples of formulas in $L(\Sigma)$. The first one a closed formula. $\triangledown$

Next, we refer to substitutions over Σ. A substitution over Σ maps terms and schema variables to terms and formulas, respectively.

Definition 6.1.8 A *substitution* over Σ is a map

$$\sigma : \Theta \cup \Xi \to T(\Sigma) \cup L(\Sigma)$$

such that $\sigma(\theta) \in T(\Sigma)$ for each $\theta \in \Theta$ and $\sigma(\xi) \in L(\Sigma)$ for each $\xi \in \Xi$. Similarly, a *ground substitution over* Σ is a map

$$\rho : \Theta \cup \Xi \to gT(\Sigma) \cup gL(\Sigma)$$

such that $\sigma(\theta) \in gT(\Sigma)$ for each $\theta \in \Theta$ and $\sigma(\xi) \in gL(\Sigma)$ for each $\xi \in \Xi$. $\triangledown$

We denote the set of all substitutions over Σ by $Sbs(\Sigma)$ and the set of all ground substitutions over Σ by $gSbs(\Sigma)$.

As expected, a substitution σ over Σ can be extended to the set of terms and formulas. It is worth noting the particular case of terms $\theta^{x'_1,\dots,x'_k}_{\theta'_1,\dots,\theta'_k}$ and formulas $\xi^{x'_1,\dots,x'_k}_{\theta'_1,\dots,\theta'_k}$.

Definition 6.1.9 Let σ be a substitution over Σ. The *extension* of σ to $T(\Sigma) \cup L(\Sigma)$ is the map

$$\hat{\sigma} : T(\Sigma) \cup L(\Sigma) \to T(\Sigma) \cup L(\Sigma)$$

inductively defined as follows

- $\hat{\sigma}(t) = t$, for $t \in Ind \cup X \cup F_0$, and $\hat{\sigma}(\theta) = \sigma(\theta)$, for $\theta \in \Theta$;
- $\hat{\sigma}(f(t_1, \dots, t_k)) = f(\hat{\sigma}(t_1), \dots, \hat{\sigma}(t_k))$, for $f \in F_k$, $t_1, \dots, t_k \in T(\Sigma)$, $k \in \mathbb{N}^+$;
- $\hat{\sigma}(\theta^{x'_1,\dots,x'_k}_{\theta'_1,\dots,\theta'_k})$ is obtained by uniformly substituting each x'_i by $\hat{\sigma}(\theta'_i)$ in the term $\hat{\sigma}(\theta)$ for each $i = 1, \dots, k$;
- $\hat{\sigma}(\varphi) = \varphi$, for $\varphi \in C_0 \cup P_0$, and $\hat{\sigma}(\xi) = \sigma(\xi)$, for $\xi \in \Xi$;
- $\hat{\sigma}(p(t_1, \dots, t_k)) = p(\hat{\sigma}(t_1), \dots, \hat{\sigma}(t_k))$, for $p \in P_k$, $t_1, \dots, t_k \in T(\Sigma)$, $k \in \mathbb{N}^+$;
- $\hat{\sigma}(t_1 \approx t_2) = (\hat{\sigma}(t_1) \approx \hat{\sigma}(t_2))$ and $\hat{\sigma}(t_1 \not\approx t_2) = (\hat{\sigma}(t_1) \not\approx \sigma(t_2))$, for $t_1, t_2 \in T(\Sigma)$;
- $\hat{\sigma}(c(\varphi_1, \dots, \varphi_k)) = (c(\hat{\sigma}(\varphi_1), \dots, \hat{\sigma}(\varphi_k)))$, for $c \in C_k$, $\varphi_1, \dots, \varphi_k \in L(\Sigma)$, $k \in \mathbb{N}^+$;
- $\hat{\sigma}((qx)(\varphi_1, \dots, \varphi_k)) = ((qx)(\hat{\sigma}(\varphi_1), \dots, \hat{\sigma}(\varphi_k)))$, for $q \in Q_k$, $x \in X$, $\varphi_1, \dots, \varphi_k \in L(\Sigma)$, $k \in \mathbb{N}^+$;
- $\hat{\sigma}(o(\varphi_1, \dots, \varphi_k)) = (o(\hat{\sigma}(\varphi_1), \dots, \hat{\sigma}(\varphi_k)))$, for $o \in O_k$, $\varphi_1, \dots, \varphi_k \in L(\Sigma)$, $k \in \mathbb{N}^+$;
- $\hat{\sigma}(\xi^{x'_1,\dots,x'_k}_{\theta'_1,\dots,\theta'_k})$ is obtained by uniformly substituting each x'_i by $\hat{\sigma}(\theta'_i)$ in the formula $\hat{\sigma}(\xi)$, for each $i = 1, \dots, k$. ∇

Example 6.1.10 Consider the signature Σ presented in Example 6.1.3 and assume again that f_1 and f_2 are function symbols of respectively arity 1 and 2. Let σ be the substitution over Σ such that $\sigma(\theta_1) = f_2(x_2, x_3)$, $\sigma(\theta_2) = f_1(x_3)$ and $\sigma(\theta_3) = f_1(x_4)$. Then

$$\hat{\sigma}({\theta_1}^{x_2}_{\theta_2}) = f_2(f_1(x_3), x_3)$$

and

$$\hat{\sigma}({\theta_1}^{x_2,x_3}_{\theta_2,\theta_3}) = f_2(f_1(x_3), f_1(x_4)).$$

Observe that we would get

$$\hat{\sigma}(\theta_1{}^{x_2,x_3}_{\theta_2,\theta_3}) = f_2(f_1(f_1(x_4)), f_1(x_4))$$

if the substitution was not uniform. ▽

In the sequel, for simplicity, we will write σ instead of $\hat{\sigma}$.

6.2 Interpretation systems

In this section we introduce the semantic aspects of first-order based logics. Since first-order languages are richer than propositional ones, the corresponding semantic structures are also more complex than the ones presented in the previous chapters for propositional based logics. Herein, we concentrate on the semantics of ground formulas. The reason for that has to do with provisos that will be discussed in Section 6.3. Also on this subject see [241].

In this setting, besides the interpretation for the connectives, we have to provide interpretations for function and predicate symbols as well as assignments to quantification variables and interpretations for quantifiers. Therefore, the semantic structures have to encompass both the semantics of modalities and the semantics of quantifiers. The main idea is to think of quantifiers as modalities.

We start by observing that, as it was done in Example 3.1.6 of Chapter 3, a Kripke frame $\langle W, R\rangle$ for a modal propositional logic induces a powerset algebra

$$\langle \wp W, [\![\cdot]\!]\rangle$$

where for each connective of arity $k > 0$, $[\![\cdot]\!]$ is a map from $(\wp W)^k$ to $\wp W$ such that:

- $[\![\neg]\!](b) = W \setminus b$;
- $[\![\Rightarrow]\!](b_1, b_2) = (W \setminus b_1) \cup b_2$;
- $[\![\Box]\!](b) = \{w \in W : \text{ if } wRw' \text{ then } w' \in b\}$.

In this way, we can look at each subset of W as a truth-value. In some cases we do not want to consider as truth-values all the subsets of W. This means that we are led to the so called general semantics [265]. A *general Kripke frame* is a tuple

$$\langle W, R, \mathcal{B}\rangle$$

where

- $\langle W, R\rangle$ is a Kripke frame;
- $\mathcal{B} \subseteq \wp W$ is such that:
  - $\mathcal{B}$ is closed for complements;

- $\mathcal{B}$ is closed for finite unions;
- $\{w \in W : wRv, \text{ for some } v \in b\} \in \mathcal{B}$ whenever $b \in \mathcal{B}$;
- $W \in \mathcal{B}$.

Again a general Kripke frame $\langle W, R, \mathcal{B}\rangle$ induces a powerset algebra

$$\langle \mathcal{B}, [\![\cdot]\!]\rangle$$

where for each connective of arity $k > 0$, $[\![\cdot]\!]$ is a map from $\mathcal{B}^k$ to $\mathcal{B}$ such that:

- $[\![\neg]\!](b) = W \setminus b$;
- $[\![\Rightarrow]\!](b_1, b_2) = (W \setminus b_1) \cup b_2$;
- $[\![\Box]\!](b) = \{w \in W : \text{ if } wRw' \text{ then } w' \in b\}$.

Another essential ingredient for understanding the semantic structures to be considered is to realize that first-order structures induce Kripke structures by looking at each $\forall x$ as a unary modal operator. Given a usual first-order structure (see, for instance, [206])

$$\langle D, [\![\cdot]\!]_F, [\![\cdot]\!]_P\rangle$$

we get, for each variable $x \in X$, a Kripke frame $\langle W, R_x\rangle$ defined as follows:

- W is D^X (the set of all assignments, that is, maps from X to D);
- $R_x \subseteq W \times W$ is such that
$$\vartheta_1 R_x \vartheta_2$$
if ϑ_1 is x-equivalent to ϑ_2, that is, $\vartheta_1(y) = \vartheta_2(y)$ for every $y \in X \setminus \{x\}$.

Hence, we consider as basic semantic units modal-like structures composed of "points" where to each point we associate an assignment and a world. We evaluate the denotation of terms and formulas at each point. The interpretation of some symbols will depend only on the assignments, while the interpretation of others will depend only on the worlds. The semantics of the quantifiers is established by looking at different points sharing the same world (by varying the assignment). On the other hand, the semantics of modalities is obtained by looking at different points sharing the same assignment (by varying the world). Hence, quantifiers can be seen as modalities with assignments playing the role of worlds.

Observe that there are some common aspects between products of modal logics in Section 1.2 of Chapter 1 and the way we define the structures above. That is, the class of structures includes the products of the structures for the modalities and the quantifiers. Observe however that we have more structures than the ones corresponding to the Cartesian product of the points.

The semantics of quantification variables is rigid, in the sense that the denotation of such variables depends only on the choice of the assignment. Hence, there is a fixed universe of individuals across the different worlds. Note, however, that we may still vary the scope of quantification from one world to another, since we do not assume that the set of assignments at a given world is composed of all functions from quantification variables to individuals.

The semantics of the connectives could have been defined regardless of assignments and worlds, but, for technical reasons related with completeness, we adopt a more general approach where their interpretation may depend on both.

Function and predicate symbols are flexible, in the sense that its interpretation can depend on the world at hand, but they are constant, in the sense that there is no dependency on the assignment at hand. The individual symbols are both rigid and constant.

We now introduce structures over Σ. For convenience, predicates will be interpreted by their characteristic maps instead of relations.

Definition 6.2.1 Let $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O\rangle$ be a signature. An *interpretation structure* over Σ is a tuple

$$\langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\,\cdot\,]\rangle$$

where

- U, $\mathbb{V}$, W and D are non-empty sets;
- $\alpha : U \to \mathbb{V}$ and $\omega : U \to W$ are maps;
- $\mathcal{E} \subseteq D^U$ and $\mathcal{B} \subseteq 2^U$ are sets such that $U \in \mathcal{B}$;
- Assume that

$$\begin{aligned} U_\vartheta &= \{u \in U : \alpha(u) = \vartheta\}, & \mathcal{B}_\vartheta &= \{b \cap U_\vartheta : b \in \mathcal{B}\},\\ U_w &= \{u \in U : \omega(u) = w\}, & \mathcal{B}_w &= \{b \cap U_w : b \in \mathcal{B}\},\\ U_{w\vartheta} &= U_w \cap U_\vartheta, & \mathcal{B}_{w\vartheta} &= \{b \cap U_{w\vartheta} : b \in \mathcal{B}\}.\end{aligned}$$

  Then $[\,\cdot\,]$ is a map defined as follows:

  – $[x] = \{[x]_\vartheta\}_{\vartheta\in\mathbb{V}}$ where $[x]_\vartheta \in D$ for $x \in X$;

  – $[\mathbf{i}] = \{[\mathbf{i}]_\vartheta\}_{\vartheta\in\mathbb{V}}$ where $[\mathbf{i}]_\vartheta \in D$ for $\mathbf{i} \in \mathit{Ind}$, and $[\mathbf{i}]_{\alpha(u)} = [\mathbf{i}]_{\alpha(u')}$ whenever $u, u' \in U_w$ for some $w \in W$;

  – $[f] = \{[f]_w\}_{w\in W}$ where $[f]_w : D^k \to D$ for $f \in F_k$;

  – $[\approx] : D^2 \to 2$ is the diagonal relation;

  – $[\not\approx] : D^2 \to 2$ is the complement of the diagonal relation;

– $[p] = \{[p]_w\}_{w\in W}$ where $[p]_w : D^k \to 2$ for $p \in P_k$;

– $[c] = \{[c]_{w\vartheta}\}_{w\in W,\,\vartheta\in\mathbb{V}}$ where $[c]_{w\vartheta} : (\mathcal{B}_{w\vartheta})^k \to \mathcal{B}_{w\vartheta}$ for $c \in C_k$;

– $[qx] = \{[qx]_w\}_{w\in W}$ where $[qx]_w : (\mathcal{B}_w)^k \to \mathcal{B}_w$ for $q \in Q_k$ and $x \in X$;

– $[o] = \{[o]_\vartheta\}_{\vartheta\in\mathbb{V}}$ where $[o]_\vartheta : (\mathcal{B}_\vartheta)^k \to \mathcal{B}_\vartheta$ for $o \in O_k$.

Moreover, the sets $\mathcal{E}$ and $\mathcal{B}$ above are assumed to be such that the following derived functions are well defined:

- $\widehat{x} : U \to \mathcal{E}$ by $\widehat{x}(u) = [x]_{\alpha(u)}$;
- $\widehat{\mathbf{i}} : U \to \mathcal{E}$ by $\widehat{\mathbf{i}}(u) = [\mathbf{i}]_{\alpha(u)}$;
- $\widehat{f} : \mathcal{E}^k \to \mathcal{E}$ by $\widehat{f}(e_1,\dots,e_k)(u) = [f]_{\omega(u)}(e_1(u),\dots,e_k(u))$;
- $\widehat{\approx} : \mathcal{E}^2 \to \mathcal{B}$ by $\widehat{\approx}(e_1,e_2)(u) = [\approx](e_1(u),e_2(u))$;
- $\widehat{\not\approx} : \mathcal{E}^2 \to \mathcal{B}$ by $\widehat{\not\approx}(e_1,e_2)(u) = [\not\approx](e_1(u),e_2(u))$;
- $\widehat{p} : \mathcal{E}^k \to \mathcal{B}$ by $\widehat{p}(e_1,\dots,e_k)(u) = [p]_{\omega(u)}(e_1(u),\dots,e_k(u))$;
- $\widehat{c} : \mathcal{B}^k \to \mathcal{B}$ by
  $\widehat{c}(b_1,\dots,b_k)(u) = [c]_{\omega(u)\alpha(u)}(b_1 \cap U_{\omega(u)\alpha(u)},\dots,b_k \cap U_{\omega(u)\alpha(u)})(u)$;
- $\widehat{qx} : \mathcal{B}^k \to \mathcal{B}$ by
  $\widehat{qx}(b_1,\dots,b_k)(u) = [qx]_{\omega(u)}(b_1 \cap U_{\omega(u)},\dots,b_k \cap U_{\omega(u)})(u)$;
- $\widehat{o} : \mathcal{B}^k \to \mathcal{B}$ by
  $\widehat{o}(b_1,\dots,b_k)(u) = [o]_{\alpha(u)}(b_1 \cap U_{\alpha(u)},\dots,b_k \cap U_{\alpha(u)})(u)$. ∇

The set U is the set of *points*, the set $\mathbb{V}$ is the set of *assignments*, the set W is the set of *worlds*, and the set D is the set of *individuals*.

Each element of $\mathcal{E}$ (a map from U to D) is an *individual concept*. Note that the denotation of the term "the president of country x" may vary with the point at hand, that is, with the assignment and the time (world) at hand.

Each element of $\mathcal{B}$ (a map from U to 2) is a *truth value*. Observe that the denotation of the formula "(the president of country x)$\approx y$" may also vary with the point at hand. From the point of view of the quantifiers and the modalities, the above structure is a powerset algebra where each set is a truth value.

Finally, the map $[\cdot]$ is the *interpretation map*. Observe that the interpretation $[x]$ depends only on the assignment ϑ at hand. The interpretation $[\mathbf{i}]$ also depends only on the assignment ϑ, but it must also be constant within a given world w. Moreover, $[f]$ and $[p]$ depend only on the world w at hand. Note that the interpretation of function symbols in F_0 and individuals $\mathbf{i}$ are different. Equality and inequality are given their usual interpretations. As already remarked above, for technical reasons, $[c]$ depends on both worlds w and assignments ϑ. With respect to the interpretation of quantifiers, $[qx]$ depends only on the world w at hand. The interpretation $[o]$ of a modality o is easily understood as the dual. It depends only on the assignment ϑ at hand.

Some comments are also in order with respect to the algebraic operations $\widehat{\cdot}$ induced by the interpretation of the symbols. A consequence of the definition of the functions $\widehat{f}$ and $\widehat{p}$ is that the denotation of formulas depends on the world at hand already at the atomic level, and not only as a result of the semantics for the modal operators.

Functions and predicates are herein dealt with as *flexible* designators since their denotations may vary across worlds. On the other hand, $\widehat{x}(u)$ does not depend on $\omega(u)$, but only on the assignment $\alpha(u)$. Hence, quantification variables are assumed to be *rigid* designators since they preserve their values across worlds. Similar comments apply to $\widehat{\mathbf{i}}(u)$.

The constraint $\widehat{\mathbf{i}}(u) = \widehat{\mathbf{i}}(u')$ whenever $u, u' \in U_w$ for some $w \in W$ imposes that individual symbols also do not change their values within a given world. For this reason we say that they are *constant* designators, besides being rigid. But note that individual symbols may still have different values in different points u and u', as long as these points are in coordinatewise disjoint subsets U_1, U_2 of the set of points, in the sense that there is no $u_1 \in U_1$ and $u_2 \in U_2$ such that $\alpha(u_1) = \alpha(u_2)$ or $\omega(u_1) = \omega(u_2)$.

Example 6.2.2 Recall the signature $\Sigma = \langle Ind, F, P, C, Q, O\rangle$ presented in Example 6.1.2. An example of an interpretation structure over Σ corresponding to classical first-order logic is the tuple

$$\langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\cdot]\rangle$$

where

- D is an non-empty set and $W = \{w\}$;
- $U = \{w\} \times \mathbb{V}$ and $\mathbb{V} = D^X$;
- $\alpha(\langle w, \vartheta\rangle) = \vartheta$ and $\omega(\langle w, \vartheta\rangle) = w$;
- $\mathcal{E} = D^U$ and $\mathcal{B} = 2^U$;
- $[x]_\vartheta = \vartheta(x)$ for $x \in X$;
- $[f] : D^k \to D$ for $f \in F_k$ with $k \geq 0$;
- $[p] : D^k \to 2$ for $p \in P_k$ with $k > 0$;
- $[\neg]_\vartheta(b) = U_\vartheta \setminus b$ for $b \in \mathcal{B}$;

- $[\Rightarrow]_\vartheta(b_1, b_2) = (U_\vartheta \setminus b_1) \cup b_2$ for $b_1, b_2 \in \mathcal{B}$;
- $[\forall x](\{w\} \times \mathbb{V}')(\langle w, \vartheta \rangle) = 1$ if $\vartheta' \in \mathbb{V}'$ for every $\vartheta' \in D^X$ such that ϑ' is x-equivalent to ϑ.

Since we only have one world, the family of functions corresponding to the interpretation of each function symbol consists of only one function. For simplicity, we identify the interpretation with that function. Similarly with respect to the interpretation of predicate symbols and quantifiers. Observe that in this example $U_w = U$ and $U_{w\vartheta} = U_\vartheta$ and the elements of $\mathcal{B}_w$ are sets $\{w\} \times \mathbb{V}'$ with $\mathbb{V}' \subseteq \mathbb{V}$. ▿

Example 6.2.3 Recall the signature $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O \rangle$ presented in Example 6.1.3. An example of an interpretation structure over Σ corresponding to modal K classical first-order logic is the tuple

$$\langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\cdot] \rangle$$

where

- D and W are non-empty sets;
- $U = W \times \mathbb{V}$ and $\mathbb{V} = D^X$;
- $\alpha(\langle w, \vartheta \rangle) = \vartheta$ and $\omega(\langle w, \vartheta \rangle) = w$;
- $\mathcal{E} = D^U$ and $\mathcal{B} = 2^U$;
- $[x]_\vartheta = \vartheta(x)$ for $x \in X$;
- $[\mathbf{i}]_\vartheta \in D$ for $\mathbf{i} \in \mathit{Ind}$;
- $[f] = \{[f]_w\}_{w \in W}$ where $[f]_w : D^k \to D$ for $f \in F_k$ with $k \geq 0$;
- $[p] = \{[p]_w\}_{w \in W}$ where $[p]_w : W \to 2$ for $p \in P_0$;
- $[p] = \{[p]_w\}_{w \in W}$ where $[p]_w : D^k \to 2$ for $p \in P_k$ with $k > 0$;
- $[\neg]_{w\vartheta}(b) = U_{w\vartheta} \setminus b$ for $b \in \mathcal{B}$;
- $[\Rightarrow]_{w\vartheta}(b_1, b_2) = (U_{w\vartheta} \setminus b_1) \cup b_2$ for $b_1, b_2 \in \mathcal{B}$;
- $[\forall x]_w(\{w\} \times \mathbb{V}')(\langle w, \vartheta \rangle) = 1$ if $\vartheta' \in \mathbb{V}'$ for every $\vartheta' \in D^X$ such that ϑ' is x-equivalent to ϑ;
- $[\Box]_\vartheta(W' \times \{\vartheta\})(\langle w, \vartheta \rangle) = 1$ if $w' \in W'$ for every $w' \in W$ such that $w R_\vartheta w'$ where $R_\vartheta \subseteq W \times W$.

Note that the elements of $\mathcal{B}_w$ and $\mathcal{B}_\vartheta$ are sets

$$(\{w\} \times \mathbb{V}') \text{ and } (W' \times \{\vartheta\})$$

with $\mathbb{V}' \subseteq \mathbb{V}$ and $W' \subseteq W$. For each world w, the function $\mathcal{I}(f)_w$ gives the corresponding interpretation of function symbol f. Similarly, the function $\mathcal{I}(p)_w$ gives the corresponding interpretation of predicate symbol p, when the arity of p is not 0. The relation R_ϑ is an accessibility relation on W. ▽

The usual choices for the sets $\mathcal{E}$ and $\mathcal{B}$ are D^U and $\wp U$, respectively, as in Examples 6.2.2 and 6.2.3. However, the above definition of interpretation structure over Σ encompasses other choices, making this structure "general" in the sense of [282]. The possibility of having other choices becomes relevant when dealing with completeness issues, as we will discuss later on.

Note that an interpretation structure over Σ has a fixed global universe D of individuals, but different domains of individuals at different worlds are allowed. Local domains are derived concepts in our case. At each world w, we should consider the following two local domains:

- $D^{\mathcal{E}}_w = \{d \in D : \text{there are } e \in \mathcal{E},\ u \in U \text{ such that } \omega(u) = w \text{ and } e(u) = d\}$;
- $D^{\mathbb{V}}_w = \{d \in D : \text{there are } x \in X,\ u \in U \text{ such that } \omega(u) = w \text{ and } [x]_{\alpha(u)} = d\}$.

The set $D^{\mathcal{E}}_w$ contains all possible denotations of terms at the world w and the set $D^{\mathbb{V}}_w$ contains all possible denotations of variables at w. Therefore, $D^{\mathbb{V}}_w$ contains all individuals which are relevant when interpreting a quantification at w. Since variables are terms, we have that $D^{\mathbb{V}}_w \subseteq D^{\mathcal{E}}_w$. In the simplest cases, we have

- $D^{\mathbb{V}}_w = D^{\mathcal{E}}_w = D$;
- $\mathbb{V}$ is isomorphic to D^X;
- $\mathcal{E} = D^U$.

This is the case in Example 6.2.3, for instance. Observe that in a logic with universal quantification, if $D^{\mathbb{V}}_w \neq D^{\mathcal{E}}_w$ then the formula

$$((\forall x\, \psi) \Rightarrow \psi^x_t)$$

can be falsified even if ψ does not contain any modality. But the formula

$$((\forall x\, \psi) \Rightarrow (E(t) \Rightarrow \psi^x_t))$$

will be valid when the existence predicate E is interpreted at each world w as $D^{\mathbb{V}}_w$ (provided that no modalities are involved).

The following notion of reduct of an interpretation structure will be useful later on when defining fibring.

Definition 6.2.4 Let Σ and Σ' be signatures such that $\Sigma \leq \Sigma'$ and let s' be an interpretation structure over Σ'. The *reduct* of s' to Σ is the interpretation structure over Σ

$$s'|_\Sigma = \langle U', \mathbb{V}', W', \alpha', \omega', D', \mathcal{E}', \mathcal{B}', [\cdot]'|_\Sigma\rangle.$$

▽

We now define the denotation of (ground) terms and formulas in the context of a given interpretation structure over Σ.

Definition 6.2.5 Given an interpretation structure $s = \langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\cdot]\rangle$ over Σ, the *denotation map*

$$[\![\cdot]\!]_s : gT(\Sigma) \cup gL(\Sigma) \to \mathcal{E} \cup \mathcal{B}$$

is inductively defined as follows:

- $[\![t]\!]_s = \widehat{t}$, for $t \in X \cup Ind$
- $[\![f(t_1, \ldots, t_k)]\!]_s = \widehat{f}([\![t_1]\!]_s, \ldots, [\![t_k]\!]_s)$,
  for $f \in F_k$ and $t_1, \ldots, t_k \in gT(\Sigma)$, $k \geq 0$;
- $[\![p(t_1, \ldots, t_k)]\!]_s = \widehat{p}([\![t_1]\!]_s, \ldots, [\![t_k]\!]_s)$
  for $p \in F_k$ and $t_1, \ldots, t_k \in gT(\Sigma)$, $k \geq 0$;
- $[\![t_1 \approx t_2]\!] = \widehat{\approx}([\![t_1]\!]_s, [\![t_2]\!]_s)$ for $t_1, t_2 \in gT(\Sigma)$;
- $[\![t_1 \not\approx t_2]\!] = \widehat{\not\approx}([\![t_1]\!]_s, [\![t_2]\!]_s)$ for $t_1, t_2 \in gT(\Sigma)$;
- $[\![c(\varphi_1, \ldots, \varphi_k)]\!]_s = \widehat{c}([\![\varphi_1]\!]_s, \ldots, [\![\varphi_k]\!]_s)$
  for $c \in C_k$ and $\varphi_1, \ldots, \varphi_k \in gL(\Sigma)$, $k \geq 0$;
- $[\![(qx)(\varphi_1, \ldots, \varphi_k)]\!]_s = \widehat{(qx)}([\![\varphi_1]\!]_s, \ldots, [\![\varphi_k]\!]_s)$
  for $q \in Q_k$, $x \in X$, and $\varphi_1, \ldots, \varphi_k \in gL(\Sigma)$, $k \geq 0$;
- $[\![o(\varphi_1, \ldots, \varphi_k)]\!]_s = \widehat{o}([\![\varphi_1]\!]_s, \ldots, [\![\varphi_k]\!]_s)$
  for $o \in O_k$ and $\varphi_1, \ldots, \varphi_k \in gL(\Sigma)$, $k \geq 0$. $\triangledown$

Next, we introduce interpretation systems and the notions of local and global entailment. An interpretation system includes a signature Σ, a class of models and a map that associates a structure over Σ to each model.

Definition 6.2.6 An *interpretation system* is a tuple

$$I = \langle \Sigma, M, A\rangle$$

where Σ is a signature, M is a class (of models) and A maps each $m \in M$ to an interpretation structure over Σ. $\triangledown$

Observe that the notion of interpretation system is not the same as in Chapter 3 where the semantic structures are ordered algebras. In this case, contrarily to what we did in previous chapters, we include explicitly the class M of models of the original logic and use the map A to indicate how to extract from each model an interpretation structure over Σ for simplicity reasons. For instance in the case of first-order logic we include in the definition of an interpretation structure over Σ the set of individuals in the original first-order model.

For simplicity, within the context of an interpretation system, we often replace $A(m)$ by m, writing for instance $[\![\cdot]\!]_m$ instead of $[\![\cdot]\!]_{A(m)}$.

Example 6.2.7 An example of interpretation system corresponding to the classical first-order logic is

$$I = \langle \Sigma, M, A \rangle$$

where

- $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O \rangle$ is as defined in Example 6.1.2.
- M is the class of all tuples of the form

  $$m = \langle D, \{w\}, V, \mathcal{I} \rangle$$

  where

  - D is a non-empty set;
  - $\mathcal{I}(f) : D^k \to D$ for $f \in F_k$, $k \geq 0$;
  - $\mathcal{I}(p) : D^k \to 2$ for $p \in P_k$, $k > 0$.
- for each model $m \in M$, the interpretation structure over Σ

  $$A(m) = \langle U, \mathbb{V}, \{w\}, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\,\cdot\,] \rangle$$

  is defined as in Example 6.2.2, considering herein

  - $[f] = \mathcal{I}(f)$, for $f \in F_k$ and $k \geq 0$;
  - $[p] = \mathcal{I}(p)$, for $p \in P_k$ and $k > 0$. ∇

Example 6.2.8 An example of interpretation system corresponding to the modal K classical first-order logic is

$$I = \langle \Sigma, M, A \rangle$$

where

- $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O \rangle$ is as defined in Example 6.1.3.

- M is the class of all tuples of the form

$$m = \langle D, W, R, V, \mathcal{I} \rangle$$

where

  - D and W are non-empty sets;
  - $R = \{R_\vartheta\}_{\vartheta \in D^X}$ with each $R_\vartheta \subseteq W \times W$;
  - $V(p) : W \to 2$ for $p \in P_0$;
  - $\mathcal{I}(\mathbf{i}) \in D$ for $\mathbf{i} \in Ind$;
  - $\mathcal{I}(f) = \{\mathcal{I}(f)_w\}_{w \in W}$ where $\mathcal{I}(f)_w : D^k \to D$ for $f \in F_k$, $k \leq 0$;
  - $\mathcal{I}(p) = \{\mathcal{I}(p)_w\}_{w \in W}$ where $\mathcal{I}(p)_w : D^k \to 2$ for $p \in P_k$, $k > 0$.

- for each model $m \in M$, the interpretation structure over Σ

$$A(m) = \langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\,\cdot\,] \rangle$$

is defined as in Example 6.2.3, considering herein

  - $[f] = \mathcal{I}(f)$, for $f \in F_k$ and $k \geq 0$;
  - $[p]_w = V(p)(w)$, for $w \in W$ and $p \in P_0$;
  - $[p] = \mathcal{I}(p)$, for $p \in P_k$ and $k > 0$.

Observe that for each $w \in W$, the pair $\langle D, \mathcal{I}_w \rangle$ is a usual first-order interpretation structure and, for each $\vartheta \in D^X$, the triple $\langle W, R_\vartheta, V \rangle$ is a Kripke model. $\triangledown$

Global and local entailment can be defined in this framework as expected.

Definition 6.2.9 Let $I = \langle \Sigma, M, A \rangle$ be an interpretation system and, for every $m \in M$, let U_m be the set of points of $A(m)$.

A formula $\varphi \in gL(\Sigma)$ is *globally entailed* by $\Gamma \subseteq gL(\Sigma)$ in I if, for every $m \in M$, $[\![\varphi]\!]_m = U_m$ whenever $[\![\gamma]\!]_m = U_m$ for every $\gamma \in \Gamma$. We will write

$$\Gamma \vDash^g_I \varphi$$

to denote that φ is globally entailed by Γ in I.

A formula $\varphi \in gL(\Sigma)$ is *locally entailed* by $\Gamma \subseteq gL(\Sigma)$ in I if, for every $m \in M$ and $u \in U_m$, $u \in [\![\varphi]\!]_m$ whenever $u \in [\![\gamma]\!]_m$ for every $\gamma \in \Gamma$. We will write

$$\Gamma \vDash^\ell_I \varphi$$

to denote that φ is locally entailed by Γ in I. $\triangledown$

We end this section with some remarks concerning the interpretation system corresponding to modal K classical first-order logic presented in Example 6.2.8. Note that the accessibility relation may depend on the assignments (but, as a particular case, we can impose $R_\vartheta = R_{\vartheta'}$ for all ϑ, ϑ' in $\mathbb{V}$). In the setting adopted herein, a modal logic with quantifiers is just a bidimensional modal logic with one dimension dedicated to the proper modality and the other dimension dedicated to the quantifiers seen as modalities over assignments. The assignment-dependent accessibility relation is essential to a achieve a semantic setting where Barcan formulas

$$((\forall x(\Box\varphi)) \Rightarrow (\Box(\forall x\,\varphi))) \quad \text{and} \quad ((\Box(\forall x\,\varphi)) \Rightarrow (\forall x(\Box\varphi)))$$

are not valid. In this aspect, our semantic construction differs from products of modal logics since no interaction is created between the quantifiers and the modal operator.

6.3 Hilbert calculi

In this section we concentrate on the deductive component of first-order based logics. We are going to consider Hilbert calculi similar to the ones introduced for propositional based logics, but where some new features have to be added.

The new features are related to the undesired interactions between binding operators and variables that often occur when substituting schema variables. For instance,

$$(\xi_1 \Rightarrow (\forall x_1\,\xi_1))$$

is a theorem of classical first-order logic, but if we substitute ξ for some formula where the variable x_1 occurs free, say $p_1(x_1)$, we get

$$(p_1(x_1) \Rightarrow (\forall x_1\,p_1(x_1)))$$

which is no longer a theorem. Hence, substitutions of schema variables have to be carefully handled when applying inference rules in a derivation. In some cases, certain substitutions should be forbidden. To ensure that only allowed substitutions are used, we add provisos to the inference rules.

Definition 6.3.1 A *proviso* over Σ is a map from $gSbs(\Sigma)$ to 2. A *proviso* is a family

$$\pi = \{\pi_\Sigma\}_{\Sigma \in \text{FOSIG}}$$

where FOSIG is the class of all first-order signatures and each π_Σ is a proviso over Σ, such that $\pi_{\Sigma'}(\rho) = \pi_\Sigma(\rho)$ for every ground substitution ρ over Σ whenever $\Sigma \leq \Sigma'$. ∇

Intuitively, we have $\pi_\Sigma(\rho) = 1$ if and only if the ground substitution ρ over Σ is allowed. The need for the definition of proviso as a family of functions indexed by signatures comes from the fact that different signatures are involved when we consider the fibring of Hilbert calculi.

We denote the sets of all provisos over Σ and all provisos by $Prov(\Sigma)$ and $Prov$, respectively. Given a proviso π we say that π_Σ is the *instance* over Σ of π. When no confusion arises we may write $\pi(\rho)$ for $\pi_\Sigma(\rho)$.

A formal treatment of provisos was first proposed in [241].

We now refer some provisos that will be often used throughout the chapter:

- the *unit proviso* **1** maps at each signature Σ every substitution over Σ to 1;
- the *zero proviso* **0** maps at each signature Σ every substitution over Σ to 0;
- for each $\xi \in \Xi$, $\theta \in \Xi$ and $x \in X$
  - $\mathrm{atm}(\xi) = \{\mathrm{atm}(\xi)_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $atm_\Sigma(\xi)(\rho) = 1$ if $\rho(\xi)$ is atomic;
  - $\mathrm{cfo}(\xi) = \{\mathrm{cfo}(\xi)_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $\mathrm{cfo}_\Sigma(\xi)(\rho) = 1$ if $\rho(\xi)$ is a closed first-order formula;
  - $\mathrm{rig}(\xi) = \{\mathrm{rig}(\xi)_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $\mathrm{rig}(\xi)_\Sigma(\rho) = 1$ if $\rho(\xi)$ is an equality or inequality of rigid terms (that is, terms in $X \cup Ind$);
  - $x \not\in \xi = \{x \not\in \xi_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $x \not\in \xi_\Sigma(\rho) = 1$ if x does not occur free in $\rho(\xi)$ and $\rho(\xi)$ does not contain modalities;
  - $\theta \rhd x{:}\xi = \{\theta \rhd x{:}\xi_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $\theta \rhd x{:}\xi_\Sigma(\rho) = 1$ if when replacing the free occurrences of x in $\rho(\xi)$ no variable in $\rho(\theta)$ is captured by a quantifier and no non-rigid replacement is made within the scope of a modality;
- for each $\xi \in \Xi$, $x \in X$ and $\Psi \subseteq L(\Sigma)$, and letting $[\xi/\psi]$ denote a substitution over Σ that replaces ξ by ψ
  - $\mathrm{cfo}(\Psi) = \{\mathrm{cfo}(\Psi)_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $\mathrm{cfo}_\Sigma(\Psi) = 1$ if $\mathrm{cfo}_\Sigma(\xi)[\xi/\psi](\rho) = 1$ for each $\psi \in \Psi$;
  - $\mathrm{rig}(\Psi) = \{\mathrm{rig}(\Psi)_\Sigma\}_{\Sigma \in \mathrm{FOSIG}}$ where $\mathrm{rig}(\Psi)_\Sigma(\rho) = 1$ if $\mathrm{rig}_\Sigma(\xi)[\xi/\psi](\rho) = 1$ for each $\psi \in \Psi$.

Example 6.3.2 Let Σ be the first-order signature presented in Example 6.1.3. Assume that p_1 and p_2 are predicate symbols of respectively arity 1 and 2, and let ρ be a ground substitution over Σ such that

- $\rho(\xi_1) = (\forall x_1\, p_1(x_1))$;
- $\rho(\xi_2) = p_1(x_1)$;
- $\rho(\xi_3) = (\forall x_1\, p_2(x_1, x_2))$.

Then,

- $\mathrm{cfo}_\Sigma(\xi_1)(\rho) = 1$ and $\mathrm{cfo}_\Sigma(\xi_2)(\rho) = 0$;
- $x_1 \not\in \xi_{3\Sigma}(\rho) = 1$ and $x_2 \not\in \xi_{3\Sigma}(\rho) = 0$. ∇

We define the *product* of two provisos $\pi = \{\pi_\Sigma\}_{\Sigma\in\mathrm{FOSIG}}$ and $\pi' = \{\pi'_\Sigma\}_{\Sigma\in\mathrm{FOSIG}}$ as the proviso

$$\pi \sqcap \pi' = \{(\pi \sqcap \pi')_\Sigma\}_{\Sigma\in\mathrm{FOSIG}}$$

where $(\pi \sqcap \pi')_\Sigma(\rho) = 1$ if $\pi_\Sigma(\rho) = 1$ and $\pi'_\Sigma(\rho) = 1$.

Given a proviso π and a substitution σ over Σ, we denote by $\pi_\Sigma\sigma$ the map such that $(\pi_\Sigma\sigma)(\rho) = \pi_\Sigma(\widehat{\rho} \circ \sigma)$. Clearly, $\mathbf{1}_\Sigma\sigma = \mathbf{1}_\Sigma$ and $\mathbf{0}_\Sigma\sigma = \mathbf{0}_\Sigma$. Moreover, for every ground substitution ρ over Σ, $\pi_\Sigma\rho = \mathbf{1}_\Sigma$ if $\pi_\Sigma(\rho) = 1$, and $\pi_\Sigma\rho = \mathbf{0}_\Sigma$ if $\pi_\Sigma(\rho) = 0$.

We now introduce (first-order based) inference rules.

Definition 6.3.3 An *inference rule* over a signature Σ is a triple

$$\langle \Psi, \varphi, \pi \rangle$$

where $\Psi \cup \varphi \subseteq L(\Sigma)$ and $\pi \in \mathit{Prov}$. ∇

Again, Ψ is the set of *premises* and φ is the *conclusion*. An inference rule is also called *axiom* if $\Psi = \emptyset$, and *rule* if $\Psi \neq \emptyset$. If Ψ is finite the inference rule is *finitary*. In the sequel, we will consider only finitary inference rules hence, for simplicity, "inference rule" will stand for "finitary inference rule".

Observe that in an inference rule $\langle \Psi, \varphi, \pi \rangle$, π is an element of *Prov* and not just *Prov*(Σ). There are technical reasons that justify this fact. In particular, we will need to consider an inference rule over Σ as an inference rule over a signature Σ' such that $\Sigma \leq \Sigma'$ and, in this situation, we need to know the value of the proviso for substitutions over Σ'.

We now introduce Hilbert calculi. We consider global rules as well as local rules. Global rules constitute the deductive counterpart of global entailment and local rules the deductive counterpart of local entailment.

Definition 6.3.4 A *first-order Hilbert calculus* is a tuple

$$\langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$$

where Σ is a signature R_ℓ, $R_{\mathrm{Q}g}$, $R_{\mathrm{O}g}$ and R_g are sets of inference rules over Σ and

- $R_\ell \subseteq R_{\mathrm{Q}g}$;

- $R_\ell \subseteq R_{\mathrm{O}g}$;
- $R_{\mathrm{Q}g} \cup R_{\mathrm{O}g} \subseteq R_g$. ∇

Each element of R_ℓ is a *local rule* while each element of $R_{\mathrm{Q}g}$ is a *quantifier global rule*, each element of $R_{\mathrm{O}g}$ is a *modal global rule* and each element of $R_{\mathrm{Q}g}$ is a *global rule*. Since in this chapter we only deal with first-order Hilbert calculi, in the sequel we will write only "Hilbert calculus" or "Hilbert calculi".

Herein, for convenience, we are not imposing that global inference rules have a non-empty set of premises. But this approach is indeed equivalent in terms of both local and global derivations.

The distinction between quantifier and modal global rules will be useful later on when defining vertically and a horizontally persistent logics. These properties are relevant when studying the preservation of completeness by fibring.

In the example below we need the notion of tautological formula. We say that a formula $\varphi \in L(\Sigma)$ is tautological when, being $\mathbb{P}$ the set of propositional symbols and C the classical (propositional) signature in Example 2.1.2 of Chapter 2, there is a map $\mu : \mathbb{P} \to L(\Sigma)$ and a formula $\psi \in gL(C)$ such that ψ is valid in $I_{\mathbf{CPL}}$ (see Example 3.1.17 of Chapter 3) and $\hat{\mu}(\psi) = \varphi$, where $\hat{\mu}$ the extension of μ to $gL(C)$ defined as expected.

Example 6.3.5 A Hilbert calculus corresponding to classical first-order logic is

$$\langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$$

where

- the signature Σ is as in Example 6.1.2;
- the inference rules in R_ℓ are the following
  - $\langle \emptyset, \varphi, \mathbf{1} \rangle$ for every tautological formula φ;
  - $\langle \emptyset, (\theta_1 \approx \theta_1), \mathbf{1} \rangle$;
  - $\langle \emptyset, ((\theta_1 \approx \theta_2) \Rightarrow (\theta_2 \approx \theta_1)), \mathbf{1} \rangle$;
  - $\langle \emptyset, ((\theta_1 \approx \theta_2) \Rightarrow ((\theta_2 \approx \theta_3) \Rightarrow (\theta_1 \approx \theta_3))), \mathbf{1} \rangle$;
  - $\langle \emptyset, ((\theta_1 \approx \theta_2) \Rightarrow (\ldots \Rightarrow ((\theta_{2k-1} \approx \theta_{2k}) \Rightarrow (\theta^{x'_1,\ldots,x'_k}_{\theta_1,\ldots,\theta_{2k-1}} \approx \theta^{x''_1,\ldots,x''_k}_{\theta_2,\ldots,\theta_{2k}})) \ldots)), \mathbf{1} \rangle$, for $\{x'_1, \ldots, x'_k, x''_1, \ldots, x''_k\} \in X$, $k \geq 1$;
  - $\langle \emptyset, ((\theta_1 \approx \theta_2) \Rightarrow (\ldots \Rightarrow ((\theta_{2k-1} \approx \theta_{2k}) \Rightarrow (\xi^{\,x'_1,\ldots,x'_k}_{\,\theta_1,\ldots,\theta_{2k-1}} \approx \xi^{\,x''_1,\ldots,x''_k}_{\,\theta_2,\ldots,\theta_{2k}})) \ldots)), \mathrm{atm}(\xi) \rangle$, for $\{x'_1, \ldots, x'_k, x''_1, \ldots, x''_k\} \in X$, $k \geq 1$;
  - $\langle \emptyset, ((\theta_1 \not\approx \theta_2) \Leftrightarrow (\neg(\theta_1 \approx \theta_2))), \mathbf{1} \rangle$;
  - $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2, \mathbf{1} \rangle$.
- the inference rules in $R_{\mathrm{Q}g}$ are the following

- all the inference rules in R_ℓ
- $\langle \emptyset, ((\forall x_1\, (\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\forall x_1\, \xi_1) \Rightarrow (\forall x_1\, \xi_2))), \mathbf{1}\rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow (\forall x_1\, \xi_1)), x_1 \notin \xi_1\rangle$;
- $\langle \emptyset, ((\forall x_1\, \xi_1) \Rightarrow {\xi_1}^{x_1}_{\theta_1}), \theta_1 \triangleright x_1{:}\xi_1\rangle$;
- $\langle \{\xi_1\}, (\forall x_1\, \xi_1), \mathbf{1}\rangle$.

* $R_{\mathrm{O}g} = R_\ell$;
* $R_g = R_{\mathrm{Q}g}$.

With respect to provisos, note that, for instance, the proviso $\theta_1 \triangleright x_1{:}\xi_1$ in axiom

$$((\forall x_1\, \xi_1) \Rightarrow {\xi_1}^{x_1}_{\theta_1})$$

is essential. Without this proviso we could infer

$$(\exists x_2\, p_2(x_2, x_2))$$

from $(\forall x_1 (\exists x_2\, p_2(x_1, x_2))))$. ▽

Example 6.3.6 A Hilbert calculus corresponding to modal K classical first-order logic is

$$\langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g\rangle$$

where

* the signature Σ is as in Example 6.1.3;
* the inference rules in R_ℓ are as in Example 6.3.5;
* the inference rules in $R_{\mathrm{Q}g}$ are as in Example 6.3.5;
* the inference rules in $R_{\mathrm{O}g}$ are the following
  - all the inference rules in R_ℓ;
  - $\langle \emptyset, ((\Box\, (\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\, \xi_1) \Rightarrow (\Box\, \xi_2))), \mathbf{1}\rangle$;
  - $\langle \emptyset, (\xi_1 \Rightarrow (\Box\, \xi_1)), \mathrm{rig}(\xi_1)\rangle$;
  - $\langle \{\xi_1\}, (\Box\, \xi_1), \mathbf{1}\rangle$.
* $R_g = R_{\mathrm{Q}g} \cup R_{\mathrm{O}g}$.

Observe that the requirement about non-rigid replacements within the scope of a modality in the proviso $\theta_1 \triangleright x_1{:}\xi_1$ is essential in axiom $((\forall x_1\, \xi_1) \Rightarrow {\xi_1}^{x_1}_{\theta_1})$, when we consider modalities. Without this requirement, given a flexible symbol s, we would be able to infer

$$((s \approx s) \Rightarrow (\Diamond(s > s)))$$

from

$$(\forall x_1 ((s \approx x_1) \Rightarrow (\Diamond(s > x)))).$$

Obviously, the latter is a satisfiable formula while the former is not. ▽

We now define the notions of global and local derivations in a Hilbert calculus.

Definition 6.3.7 Let $H = \langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ be a Hilbert calculus. A formula $\varphi \in L(\Sigma)$ is *globally derivable* from the set $\Gamma \subseteq L(\Sigma)$ in H *with proviso* $\pi \in \mathit{Prov}(\Sigma)$ if there is a finite sequence

$$\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$$

of pairs in $L(\Sigma) \times \mathit{Prov}(\Sigma)$ such that

- $\pi \neq \mathbf{0}_\Sigma$;
- φ is φ_n and π is π_n;
- for each $i = 1, \ldots, n$, either $\varphi_i \in \Gamma$ and $\pi_i = \mathbf{1}_\Sigma$, or there are a substitution σ over Σ and a rule $\langle \Psi, \psi, \pi' \rangle \in R_g$ such that
  - φ_i is $\sigma(\psi)$;
  - $\sigma(\Psi) = \{\psi_{j_1}, \ldots, \psi_{j_k}\} \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$;
  - $\pi_i = \pi_{j_1} \sqcap \ldots \sqcap \pi_{j_k} \sqcap \pi'_\Sigma \sigma$.

The sequence $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$ is a *global derivation* of φ from Γ in H *with proviso* π. We write

$$\Gamma \vdash^g_H \varphi \lhd \pi$$

to denote that φ is globally derivable from Γ in H with proviso π. ▽

If $\pi = \mathbf{1}_\Sigma$ we may omit the proviso π. Whenever $\emptyset \vdash^g_H \varphi \lhd \pi$ we say that φ is a *theorem with proviso* π, or just *theorem* if $\pi = \mathbf{1}_\Sigma$.

Definition 6.3.8 Let $H = \langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ be a Hilbert calculus. A formula $\varphi \in L(\Sigma)$ is *locally derivable* from the set $\Gamma \subseteq L(\Sigma)$ in H *with proviso* $\pi \in \mathit{Prov}(\Sigma)$ if there is a finite sequence

$$\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$$

of pairs in $L(\Sigma) \times \mathit{Prov}(\Sigma)$ such that

- $\pi \neq \mathbf{0}_\Sigma$;
- φ is φ_n and π is π_n;
- for each $i = 1, \ldots, n$, either $\varphi_i \in \Gamma$ and $\pi_i = \mathbf{1}_\Sigma$, or φ_i is a theorem with proviso π' and $\pi_i = \pi'$, or there are a substitution σ over Σ and a rule $\langle \Psi, \psi, \pi' \rangle \in R_\ell$ and such that
  - φ_i is $\sigma(\psi)$;
  - $\sigma(\Psi) = \{\psi_{j_1}, \ldots, \psi_{j_k}\} \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$;

– $\pi_i = \pi_{j_1} \sqcap \ldots \sqcap \pi_{j_k} \sqcap \pi'_\Sigma \sigma$.

The sequence $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$ is a *local derivation* of φ from Γ in H *with proviso* π. We write

$$\Gamma \vdash^{\ell}_{H} \varphi \lhd \pi$$

to denote that φ is locally derivable from Γ in H with proviso π. $\triangledown$

Besides global and local derivations, in this first-order setting we also consider two other kinds of derivations: global Q-derivations and global O-derivations. A Q-derivation is similar to a local derivation, but we allow the use of some global rules, the global quantifier rules. The case of O-derivations is analogous.

Definition 6.3.9 Let $H = \langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ be a Hilbert calculus. A formula $\varphi \in L(\Sigma)$ is *Q-globally derivable* from $\Gamma \subseteq L(\Sigma)$ in H *with proviso* $\pi \in \mathit{Prov}(\Sigma)$ if there is a finite sequence

$$\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$$

of pairs in $L(\Sigma) \times \mathit{Prov}(\Sigma)$ verifying all the conditions described in Definition 6.3.8, but where the inference rules allowed are the inference rules in $R_{\mathrm{Q}g}$.

The sequence $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$ is a *Q-global derivation* of φ from Γ in H *with proviso* π. We write

$$\Gamma \vdash^{g}_{\mathrm{Q}H} \varphi \lhd \pi$$

to denote that φ is Q-globally derivable from Γ in H with proviso π. $\triangledown$

Definition 6.3.10 Let $H = \langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ be a Hilbert calculus. A formula $\varphi \in L(\Sigma)$ is *O-globally derivable* from $\Gamma \subseteq L(\Sigma)$ in H *with proviso* $\pi \in \mathit{Prov}(\Sigma)$ if there is a finite sequence

$$\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$$

of pairs in $L(\Sigma) \times \mathit{Prov}(\Sigma)$ verifying all the conditions described in Definition 6.3.8, but where the inference rules allowed are the inference rules in $R_{\mathrm{O}g}$.

The sequence $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$ is a *O-global derivation* of φ from Γ in H *with proviso* π. We write

$$\Gamma \vdash^{g}_{\mathrm{O}H} \varphi \lhd \pi$$

to denote that φ is O-globally derivable from Γ in H with proviso π. $\triangledown$

As in the case of global and local derivations, the proviso can be omitted whenever it is the unit proviso.

Example 6.3.11 Let H be the Hilbert calculus presented in Example 6.3.6. Assume that p_1 is a predicate symbol of arity 1. The following is a global derivation of $(\Box p_1(x_2))$ from $(\forall x_1\, p_1(x_1))$ in H:

1. $\langle(\forall x_1\, p_1(x_1)), \mathbf{1}_\Sigma\rangle$
2. $\langle(((\forall x_1\, p_1(x_1)) \Rightarrow p_1(x_2)), \mathbf{1}_\Sigma\rangle$
3. $\langle p_1(x_2), \mathbf{1}_\Sigma\rangle$
4. $\langle(\Box p_1(x_2)), \mathbf{1}_\Sigma\rangle$

From the above derivation we conclude that

$$(\forall x_1\, p_1(x_1)) \vdash^g_H (\Box p_1(x_2)).$$

Step 1 in the above derivation corresponds to the hypothesis $(\forall x_1\, p_1(x_1))$.

To obtain 2, we use the axiom

$$\langle\emptyset, ((\forall x_1\, \xi_1) \Rightarrow {\xi_1}^{x_1}_{\theta_1}), \theta_1 \triangleright x_1{:}\xi_1\rangle$$

and the ground substitution ρ over Σ such that $\rho(\xi_1) = p_1(x_1)$ and $\rho(\theta_1) = x_2$. Note that the corresponding proviso, π_2, is $\mathbf{1}_\Sigma$ since $\pi_2 = \theta_1 \triangleright x_1{:}\xi_{1\Sigma}\rho$, by the definition of derivation, and $\theta_1 \triangleright x_1{:}\xi_{1\Sigma}\rho = \mathbf{1}_\Sigma$ because $\theta_1 \triangleright x_1{:}\xi_{1\Sigma}(\rho) = 1$.

To obtain 3 we use 1, 2 and the modus ponens rule

$$\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2, \mathbf{1}\rangle$$

together with the ground substitution ρ' such that $\rho'(\xi_1) = (\forall x_1\, p_1(x_1))$ and $\rho'(\xi_2) = p_1(x_2)$. The proviso π_3 is $\mathbf{1}_\Sigma \sqcap \mathbf{1}_\Sigma \sqcap \mathbf{1}_\Sigma\rho'$, hence, $\pi_3 = \mathbf{1}_\Sigma$.

Finally, step 4 is obtained from 3 using the necessitation rule

$$\langle\{\xi_1\}, (\Box\, \xi_1), \mathbf{1}\rangle$$

and the ground substitution ρ'' such that $\rho''(\xi_1) = p_1(x_2)$. The proviso π_4 is $\mathbf{1}_\Sigma \sqcap \mathbf{1}_\Sigma\rho''$, hence, $\pi_4 = \mathbf{1}_\Sigma$. ∇

We denote by $\Gamma^{\vdash^g_H}$ the set of all formulas globally derivable from Γ in H with the unit proviso, and say that Γ is *globally closed in H* if $\Gamma^{\vdash^g_H} = \Gamma$. In a similar way, we can define the notions of *locally closed*, *Q-globally closed* and *O-globally closed* sets and use a similar notation.

As a result of the inclusion relationships between the sets of rules in a Hilbert calculus H, we can established some relationships between the different kinds of derivations presented above:

- if $\Gamma \vdash^\ell_H \varphi \lhd \pi$ then $\Gamma \vdash^g_{\mathrm{Q}H} \varphi \lhd \pi$ and $\Gamma \vdash^g_{\mathrm{O}H} \varphi$;
- if $\Gamma \vdash^g_{\mathrm{Q}H} \varphi \lhd \pi$ or $\Gamma \vdash^g_{\mathrm{O}H} \varphi \lhd \pi$ then $\Gamma \vdash^g_H \varphi \lhd \pi$.

Furthermore, every theorem φ is Q-globally derivable and O-globally derivable from any set Γ.

A global derivation of φ from Γ in H with proviso π is said to be *sober* whenever no proper subsequence of this derivation is also a global derivation of φ from Γ

in H with proviso π. *Sober* local derivations, Q-global derivations and O-global derivations are defined in the same way. Clearly, from any such derivations we can always extract a sober one by removing superfluous elements.

If a deduction is done without using schema variables, the resulting π is necessarily the unit proviso $\mathbf{1}_\Sigma$. Another way of obtaining such non schematic results is by producing an instance of a schematic result by applying a ground substitution $\rho \in gSbs(\Sigma)$ such that $\pi(\rho) = 1$.

With respect to non schematic deductions, we have the following result.

Proposition 6.3.12 *Let $H = \langle \Sigma, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ be a Hilbert calculus and assume that*

$$\langle \varphi_1, \pi_1 \rangle \dots \langle \varphi_n, \pi_n \rangle$$

is a sober global derivation of φ from Γ in H with proviso π. If $\rho \in gSbs(\Sigma)$ and $\pi_\Sigma(\rho) = 1$ then

$$\langle \rho(\varphi_1), \mathbf{1}_\Sigma \rangle \dots \langle \rho(\varphi_n), \mathbf{1}_\Sigma \rangle$$

is a global derivation of $\rho(\varphi)$ from $\rho(\Gamma)$ in H.

Similarly with respect to local derivations, Q-global derivations and O-global derivations.

Proof. The result is easily established using induction. $\triangleleft$

Proposition 6.3.14 states sufficient conditions for replacing individual symbols by variables in a local derivation. This property will be used later on, when studying the preservation by fibring of some properties of Hilbert calculi. We use φ^i_x to denote the formula we obtain by substituting the occurrences of the individual symbol i for x in the formula φ. Similarly, given a set Γ of formulas, $\Gamma^i_x = \{\varphi^i_x : \varphi \in \Gamma\}$.

Definition 6.3.13 A Hilbert calculus $H = \langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ is said to be *uniform* if, for every rule $\langle \Psi, \varphi, \pi \rangle \in R_g$,

$$\pi_\Sigma(\rho) = \pi_\Sigma(\rho')$$

where, for each $\theta \in \Theta$ and $\xi \in \Xi$, $\rho'(\theta)$ and $\rho'(\xi)$ are respectively obtained from $\rho(\theta)$ and $\rho(\xi)$ by replacing some occurrences of an individual symbol i by x, provided that x does not occur in $\rho(\Gamma) \cup \{\rho(\varphi)\}$. $\triangledown$

Proposition 6.3.14 *Let $H = \langle \Sigma, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ be a uniform Hilbert calculus. If*

$$\Gamma \vdash^\ell_H \varphi \triangleleft \pi$$

and x is variable not occurring in a corresponding local derivation of φ from Γ, then

$$\Gamma^{\mathbf{i}}_x \vdash^\ell_H \varphi^{\mathbf{i}}_x \triangleleft \pi$$

for every individual symbol $\mathbf{i}$ not occurring in the rules of H.

Proof. Let $r = \langle \Psi, \varphi, \pi \rangle$ be a rule in H, $\mathbf{i}$ an individual symbol not occurring in Ψ, x a variable not occurring in $\Psi \cup \{\varphi\}$, and σ a substitution over Σ. Then, if $\{\varphi_1, \ldots, \varphi_k\} \subseteq \sigma(\Psi)$ then

$$\{(\varphi_1)_x^{\mathbf{i}}, \ldots, (\varphi_k)_x^{\mathbf{i}}\} \subseteq \sigma(\Psi)_x^{\mathbf{i}}.$$

Therefore, if ψ is a theorem, then a global derivation of ψ in H can be turned to global derivation of $\psi_x^{\mathbf{i}}$ in H by replacing every substitution σ involved in the derivation by $\sigma_x^{\mathbf{i}}$. Moreover, it is trivial that $\psi \in \Gamma$ implies $\psi_x^{\mathbf{i}} \in \Gamma_x^{\mathbf{i}}$. Then, forgetting the provisos for the moment, the replacement of substitutions σ by $\sigma_x^{\mathbf{i}}$ transform a local derivation of φ from Γ into a local derivation of $\varphi_x^{\mathbf{i}}$ from $\Gamma_x^{\mathbf{i}}$.

As far as the provisos are concerned,

$$\pi_\Sigma \sigma_x^{\mathbf{i}}(\rho) = \pi_\Sigma(\hat{\rho} \circ \sigma_x^{\mathbf{i}}) = \pi_\Sigma(\hat{\rho} \circ \sigma) = \pi_\Sigma \sigma(\rho)$$

for every ρ. Hence, $\pi_\Sigma \sigma_x^{\mathbf{i}} = \pi_\Sigma \sigma$. This shows that the replacement of substitutions σ by $\sigma_x^{\mathbf{i}}$ do not change the constraints in the derivation. ◁

6.4 First-order logic systems

In this section we present first-order logic systems. As before, a logic system include both deductive and semantic aspects of a logic and constitute the right setting to state soundness and completeness properties.

Definition 6.4.1 A *first-order logic system* is a tuple

$$\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$$

such that $\langle \Sigma, M, A \rangle$ is an interpretation system and $\langle \Sigma, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ is a Hilbert calculus. ▽

Given a first-order logic system $\mathcal{L}$ we denote by $I(\mathcal{L})$ and $H(\mathcal{L})$ the underlying interpretation system and Hilbert system respectively. We say that φ is globally derivable from Γ in $\mathcal{L}$, and write

$$\Gamma \vdash_{\mathcal{L}}^{g} \varphi$$

whenever $\Gamma \vdash_{H(\mathcal{L})}^{g} \varphi$. Similarly with respect to Q-globally derivable, O-globally derivable and locally derivable formulas. Moreover, we say that φ is globally entailed from Γ in $\mathcal{L}$, and write

$$\Gamma \vDash_{\mathcal{L}}^{g} \varphi$$

whenever $\Gamma \vDash_{I(\mathcal{L})}^{g} \varphi$. Similarly, with respect to locally entailed formulas.

In the sequel, for simplicity, we will often use "logic system" instead of "first-order logic system".

Example 6.4.2 A first-order logic system corresponding to classical first-order logic is $\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ where

- $\langle \Sigma, M, A \rangle$ is the interpretation system presented in Example 6.2.7;
- $\langle \Sigma, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ is the Hilbert calculus presented in Example 6.3.5. $\triangledown$

Example 6.4.3 A first-order logic system corresponding to modal K classical first-order logic is $\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ where

- $\langle \Sigma, M, A \rangle$ is the interpretation system presented in Example 6.2.8;
- $\langle \Sigma, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ is the Hilbert calculus presented in Example 6.3.6. $\triangledown$

We now concentrate on the properties of soundness and completeness.

Definition 6.4.4 Let $\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ be a first-order logic system. Then, $\mathcal{L}$ is

- *globally sound* if $\Gamma \vDash^g_{\mathcal{L}} \varphi$ whenever $\Gamma \vdash^g_{\mathcal{L}} \varphi$ for every $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$;
- *locally sound* if $\Gamma \vDash^\ell_{\mathcal{L}} \varphi$ whenever $\Gamma \vdash_{\mathcal{L}} \varphi$ for every $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$;
- *globally complete* if $\Gamma \vdash^g_{\mathcal{L}} \varphi$ whenever $\Gamma \vDash^g_{\mathcal{L}} \varphi$ for every $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$;
- *locally complete* if $\Gamma \vdash^\ell_{\mathcal{L}} \varphi$ whenever $\Gamma \vDash^\ell_{\mathcal{L}} \varphi$ for every $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$.

The logic system $\mathcal{L}$ is said to be *sound* if it is *globally sound* and *locally sound* and *complete* if it is *globally complete* and *locally complete*. $\triangledown$

Observe that soundness and completeness are stated only for ground formulas. Indeed, it would be impossible to consider those notions for formulas with schema variables since there is no semantic counterpart to provisos.

Definition 6.4.5 Let s be a interpretation structure over Σ with set of points U. This interpretation structure over Σ is said to be *appropriate* for a Hilbert calculus $H = \langle \Sigma, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ if

- for every $\langle \Psi, \varphi, \pi \rangle \in R_g$ and ground substitution ρ over Σ, $[\![\rho(\varphi)]\!]_s = U$ whenever $[\![\rho(\psi)]\!]_s = U$ for every $\psi \in \Psi$ and $\pi(\rho) = 1$;
- for every $\langle \Psi, \eta, \pi \rangle \in R_\ell$, ground substitution ρ over Σ and $u \in U$, $u \in [\![\rho(\varphi)]\!]_s$ whenever $u \in [\![\rho(\psi)]\!]_s$ for every $\psi \in \Psi$ and $\pi(\rho) = 1$. $\triangledown$

Proposition 6.4.6 *The logic system $\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{Qg}, R_{Og}, R_g \rangle$ is sound if and only if $A(m)$ is appropriate for $H(\mathcal{L})$ for every $m \in M$.*

The last result is the equivalent of Proposition 3.3.7 in Chapter 3 observing that appropriateness corresponds to soundness of inference rules.

Example 6.4.7 It is easy to check that the logic systems introduced in Examples 6.4.2 and 6.4.3 are sound. ▽

In general, the completeness property is not easy to establish. In Subsection 6.6.2 we present a completeness theorem for first-order logic systems. This result will play an important role when establishing preservation of completeness by fibring.

As in the propositional case (see Subsection 3.3.3 in Chapter 3), the notion of full logic system will also play an important role when studying the preservation of some properties of logic systems by fibring.

Definition 6.4.8 A logic system $\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$ is said to be *full* if for every interpretation structure s over Σ appropriate for the Hilbert system $H(\mathcal{L})$ there is a model $m \in M$ such that $A(m) = s$. ▽

Example 6.4.9 The logic system presented in Example 6.4.3 is not full. But, we can enrich it with all interpretation structures over Σ appropriate for the underlying Hilbert calculus obtaining a full logic system for modal K classical first-order logic. ▽

6.5 Fibring

In this section we define fibring of first-order based logics. The basic ideas about fibring in Chapter 3 still hold in this case. The result of fibring two logic systems $\mathcal{L}'$ and $\mathcal{L}''$ is a new logic system $\mathcal{L}$ whose signature is the union $\Sigma' \cup \Sigma''$ of the signatures of the two given logic systems.

The interpretation structures for $\Sigma' \cup \Sigma''$ should provide denotations for the symbols in this signature. The reducts of those interpretation structures to Σ' should correspond to interpretation structures in $\mathcal{L}'$ for Σ'. Similarly with respect to their reducts to Σ''. There is only a small detail about this. We only consider appropriate structures in the fibring because in this case soundness is more delicate because of the provisos in the rules.

The sets of rules of $H(\mathcal{L})$ are the unions of the corresponding sets of rules of the underlying Hilbert calculi of $\mathcal{L}'$ and $\mathcal{L}''$.

To simplify the treatment of fibring we are going to consider herein only sound logic systems. This is a reasonable assumption, given that the interesting logic systems are usually sound.

Definition 6.5.1 Given two sound first-order logic systems

$$\mathcal{L}' = \langle \Sigma', M', A', {R_\ell}', R'_{\mathrm{Q}g}, R'_{\mathrm{O}g}, {R_g}' \rangle$$

and

$$\mathcal{L}'' = \langle \Sigma'', M'', A'', R_\ell{}'', R''_{\mathrm{Q}g}, R''_{\mathrm{O}g}, R_g{}'' \rangle$$

their *fibring* is the logic system

$$\mathcal{L}' \cup \mathcal{L}'' = \langle \Sigma, M, A, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$$

where:

- $\Sigma = \Sigma' \cup \Sigma''$;
- M is the class of all interpretation structures s over Σ such that
  - $s|_{\Sigma'} \in A'(M')$ and $s|_{\Sigma''} \in A''(M'')$;
  - s is appropriate for $H(\mathcal{L}' \cup \mathcal{L}'')$;
- $A(s) = s$ for each $s \in M$;
- $R_\ell = R_\ell{}' \cup R_\ell{}''$ and $R_g = R_g{}' \cup R_g{}''$;
- $R_{\mathrm{Q}g} = R'_{\mathrm{Q}g} \cup R''_{\mathrm{Q}g}$, $R_{\mathrm{O}g} = R'_{\mathrm{O}g} \cup R''_{\mathrm{O}g}$. ∇

As before, the symbols in $\Sigma' \cap \Sigma''$ are said to be *shared*. If no symbols are shared we say that the fibring is *unconstrained*. Otherwise, we say that it is *constrained*.

Observe that, in the above definition of the class M, each interpretation structure s is required to be appropriate for the underlying Hilbert calculus $H(\mathcal{L}' \cup \mathcal{L}'')$. This requirement is essential to ensure that appropriateness, and therefore soundness, are properties preserved by fibring. It may happen, in fact, that $s|_{\Sigma'}$ is appropriate for a rule r' in $H(\mathcal{L}')$, but s is not appropriate for r' in $H(\mathcal{L}' \cup \mathcal{L}'')$: in the richer language there can be new instances of r'. An example of this situation is the usual first-order axiom

$$(\xi \Rightarrow (\forall x\, \xi))$$

where x is not free in ξ, which can be falsified if the language contains modalities.

Note that each rule $\langle \Psi, \varphi, \pi \rangle$ in $H(\mathcal{L}')$ is also a rule over the signature $\Sigma' \cup \Sigma''$. On one hand $\Sigma' \leq \Sigma' \cup \Sigma''$ and therefore $\Psi \cup \{\varphi\} \subseteq L(\Sigma' \cup \Sigma'')$. On the other hand, recall that in an inference rule over a signature provisos are elements of *Prov*, no matter the signature at hand. Hence, the proviso π in the rule is still a suitable proviso for an inference rule over $\Sigma' \cup \Sigma''$. Inference rules provisos have a component for each possible first-order signature. Hence, besides providing the requirements on ground substitutions for derivations in $H(\mathcal{L}')$, π also provides the corresponding requirements on ground substitution over $\Sigma' \cup \Sigma''$ for derivations in $H(\mathcal{L}' \cup \mathcal{L}'')$. The relevant component of π in these derivations is

$$\pi_{\Sigma' \cup \Sigma''}.$$

Similar comments also apply to rules in $H(\mathcal{L}')$.

We say that a logic system is *uniform* if the underlying Hilbert calculus is uniform.

We now illustrate the fibring construction considering first the simple example of generating a bimodal first-order logic by fibring two unimodal first-order logics.

Example 6.5.2 Let KFOL$'$ and KFOL$''$ be two copies of the full logic system for modal K classic first-order logic referred in Example 6.4.9 such that Σ' is identical to Σ'' with the exception that $O'_1 = \{\Box'\}$ and $O''_1 = \{\Box''\}$. Then,

- in the signature of the fibring KFOL$'\cup$KFOL$''$ the connectives $\neg$ and $\Rightarrow$ are shared, as well as the quantifier $\forall$, but there are two modalities: $\Box'$ and $\Box''$;
- each model in KFOL$'\cup$KFOL$''$ is an interpretation structure whose reducts are interpretation structures corresponding to models in the corresponding logic systems;
- the sets of rules in KFOL$'\cup$KFOL$''$ are obtained by the union of the corresponding sets of rules in the given logic systems. ∇

We now consider a more complex example where we obtain modal K classical first-order logic as the fibring of pure first-order logic and modal logic enriched with variables, individual symbols, equality and inequality.

Example 6.5.3 The goal is to obtain modal K classical first-order logic by fibring classical first-order logic and the K modal propositional logic described below.

The logic system FOL, corresponding to classical first-order logic is such that

- the signature Σ' is defined as in Example 6.1.2;
- the interpretation system $I(\text{FOL}) = \langle \Sigma', M', A' \rangle$ is such that M' is the class of all interpretation structures over Σ' appropriate for $H(\text{FOL})$ and A' is the identity map;
- the Hilbert calculus $H(\text{FOL})$ is defined as in Example 6.3.5.

Observe that FOL is therefore a full logic system.

When defining a modal propositional logic as a first-order based logic we are required to include in the language variables, as well as equalities and inequalities between them. So, we obtain a richer modal logic that nevertheless is quite appropriate to the intended goal. In the richer modal logic, the entailments are the same for the original formulas. Moreover, it is easy to establish a complete axiomatic system for the richer modal logic, given a complete axiomatic system for the original modal logic.

The first-order based modal logic system KML$^+$ is defined as follows

- the signature $\Sigma'' = \langle Ind'', F'', P'', C'', Q'', O'' \rangle$ is such that
  - $F' = \emptyset$ for every $k \in \mathbb{N}$;
  - P''_0 is a countable set and $P''_k = \emptyset$ for every $k \in \mathbb{N}^+$;
  - $C''_1 = \{\neg\}$, $C''_2 = \{\Rightarrow\}$ and $C''_k = \emptyset$ for $k \in \mathbb{N}\backslash\{1, 2\}$;

- $Q''_k = \emptyset$ for every $k \in \mathbb{N}^+$;
- $O''_1 = \{\Box\}$ and $O''_k = \emptyset$ for every $k \in \mathbb{N}\backslash\{1\}$;

- the interpretation system $I(\text{KML}^+) = \langle \Sigma'', M'', A'' \rangle$ is such that M'' is the class of all interpretation structures over Σ'' appropriate for the Hilbert calculus $H(\text{KML}^+)$ and A'' is the identity map;
- the Hilbert calculus $H(\text{KML}^+) = \langle \Sigma'', R_\ell{}'', R''_{\text{Q}g}, R''_{\text{O}g}, R_g{}'' \rangle$ is such that $R_\ell{}''$ and $R''_{\text{O}g}$ are defined as in Example 6.3.6, $R''_{\text{Q}g} = R_\ell{}''$ and $R_g{}'' = R''_{\text{O}g}$.

Observe that KML^+ is also a full logic system.

Finally, we obtain the logic system KFOL by fibring FOL and KML^+, that is

$$\text{KFOL} = \text{FOL} \cup \text{KML}^+.$$

The signature of the fibring $\text{KFOL} = \text{FOL} \cup \text{KML}^+$ is $\Sigma' \cup \Sigma''$ where the connectives $\neg$ and $\Rightarrow$ are shared. Note how important it was to endow the logic systems with a full semantics in order to obtain the envisaged models in the fibring. Otherwise, in the fibring, the modal part might collapse into classical logic. ∇

Example 6.5.4 Consider the logic system $\mathcal{L} = \text{KFOL}' \cup \text{KFOL}''$ referred in Example 6.5.2. The following is a global derivation for

$$(\forall x_1\, p_1(x_1)) \vdash^g_{H(\mathcal{L})} (\Box''(\Box' p_1(x_2)))$$

where Σ is the signature of $\mathcal{L}$:

1. $\langle (\forall x_1\, p_1(x_1)), \mathbf{1}_\Sigma \rangle$
2. $\langle ((\forall x_1\, p_1(x_1)) \Rightarrow p_1(x_2)), \mathbf{1}_\Sigma \rangle$
3. $\langle p_1(x_2), \mathbf{1}_\Sigma \rangle$
4. $\langle (\Box' p_1(x_2)), \mathbf{1}_\Sigma \rangle$
5. $\langle (\Box''(\Box' p_1(x_2))), \mathbf{1}_\Sigma \rangle$

Steps 1 to 4 are similar to the corresponding steps in the derivation presented in Example 6.3.11. Step 5 uses the necessitation rule $\langle \{\xi_1\}, (\Box''\, \xi_1), \mathbf{1} \rangle$. ∇

We conclude this section with a result comparing derivations in the given logics with derivations in the fibring. Given a derivation $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$, for instance in $\mathcal{L}'$, of φ from Γ constrained by π, precisely the same sequence $\langle \varphi_1, \pi_1 \rangle \ldots \langle \varphi_n, \pi_n \rangle$ constitutes a derivation in $\mathcal{L}$ of φ from Γ constrained by π. Thus we have the following proposition.

Proposition 6.5.5 *In the fibring logic system $\mathcal{L} = \mathcal{L}' \cup \mathcal{L}''$, if either $\Gamma \vdash^{\ell}_{H(\mathcal{L}')} \varphi \lhd \pi$ or $\Gamma \vdash^{\ell}_{H(\mathcal{L}'')} \varphi \lhd \pi$, then*

$$\Gamma \vdash^{\ell}_{H(\mathcal{L})} \varphi \lhd \pi.$$

∇

Hence, the resulting Hilbert calculus $H(\mathcal{L}' \cup \mathcal{L}'')$ is an extension of the two given Hilbert calculus $H(\mathcal{L}')$ and $H(\mathcal{L}'')$, that is

$$H(\mathcal{L}') \leq H(\mathcal{L}) \text{ and } H(\mathcal{L}'') \leq H(\mathcal{L})$$

meaning that the fibring is stronger than the original logic systems.

We synthesize the properties of the fibring of first-order based logics in the following way:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *homogeneous combination mechanism at the semantic level*: both original logics are presented by interpretation structures;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fibring;
- *algorithmic combination of logics at the semantic level*: given the classes of interpretation structures for the original logics, we know how to define the class of interpretation structures for the fibring, but in many cases the given logics have to be pre-processed (that is, the interpretation structures for the original logics have to be extracted).

6.6 Preservation results

In this section we present some properties of logic systems and their underlying Hilbert calculi, and discuss their preservation by fibring. Some of these properties were already given in Subsection 2.3.2 of Chapter 2. Herein, we recast them in terms of first-order based signatures.

6.6.1 Metatheorems

Herein we introduce several interesting properties of Hilbert calculi and present sufficient conditions for their preservation by fibring. These properties, and their preservation by fibring, are relevant to the study of preservation of completeness in Subsection 6.6.2.

We first refer to Q-globally persistent and O-globally persistent Hilbert calculus.

Definition 6.6.1 We say that a Hilbert calculus is *Q-globally persistent* if, for every rule $r = \langle \Psi, \varphi, \pi \rangle \in R_{\mathrm{Q}g} \setminus R_\ell$, either $\Psi = \emptyset$, or

- $\Psi = \{\psi\}$;
- φ is of the form $(op(\psi))$;
- $\psi \vdash_H^\ell (op(\psi)) \lhd \mathrm{cfo}(\{\psi\})$;
- $\{(op(\psi_1)), \ldots, (op(\psi_k))\} \vdash_H^\ell (op(\varphi')) \lhd \pi'$ for every inference rule

  $$\langle \{\psi_1, \ldots, \psi_k\}, \varphi' \rangle$$

  in R_ℓ with $k > 0$;

where op is a connective, a quantifier or a modality.

A *O-globally persistent* Hilbert calculus is defined in the same way, by replacing $R_{\mathrm{Q}g}$ by $R_{\mathrm{O}g}$, and $\mathrm{cfo}(\{\psi\})$ by $\mathrm{rig}(\{\psi\})$.

A logic system $\mathcal{L}$ is said to be *Q-globally persistent* or *O-globally persistent* whenever so is the underlying Hilbert calculus. $\triangledown$

The properties of Q-global persistence and of O-global persistence are generalizations of usual properties of first-order and of modal logics. For instance, the first-order rule of generalization has the properties referred in Definition 6.6.1, where $(op(\psi))$ is of course $(\forall x\, \psi)$. In particular:

$$\{(\forall x\, \varphi_1), (\forall x(\varphi_1 \Rightarrow \varphi_2))\} \vdash_H^\ell (\forall x\, \varphi_2).$$

We now establish a preservation result for Q-global persistence and O-global persistence.

Theorem 6.6.2 *The fibring of Q-global persistent and O-global persistent Hilbert calculi is also a Q-global persistent and O-global persistent Hilbert calculus.*

Proof. The first two properties for Q-global persistence and of O-global persistence hold in the fibring by Definition 6.5.1. The preservation of the two last properties is a consequence of Proposition 6.5.5. $\lhd$

Next we refer to vertically and horizontally persistent Hilbert calculi. These notions shows the need for the distinction between quantifier and modal inference rules.

Definition 6.6.3 The Hilbert calculus H is said to be *vertically persistent* whenever

$$\frac{\Gamma^{\vdash^g_H}, \Gamma' \vdash^g_{\mathrm{Q}H} \varphi \lhd \pi \sqcap \mathrm{cfo}_\Sigma(\Gamma')}{\Gamma^{\vdash^g_H}, \Gamma' \vdash^\ell_H \varphi \lhd \pi \sqcap \mathrm{cfo}_\Sigma(\Gamma')} \qquad (VP)$$

for every $\Gamma, \Gamma' \subseteq L(\Sigma)$ and $\varphi \in L(\Sigma)$, and it is said to be *horizontally persistent* whenever

$$\frac{\Gamma^{\vdash^g_H}, \Gamma' \vdash^g_{\mathrm{O}H} \varphi \lhd \pi \sqcap \mathrm{rig}_\Sigma(\Gamma')}{\Gamma^{\vdash^g_H}, \Gamma' \vdash^\ell_H \varphi \lhd \pi \sqcap \mathrm{rig}_\Sigma(\Gamma')} \qquad (HP)$$

for every $\Gamma, \Gamma' \subseteq L(\Sigma)$ and $\varphi \in L(\Sigma)$.

The Hilbert calculus H is said to be *persistent* if it is both vertically and horizontally persistent. A logic system is said to be *persistent* whenever so is the underlying Hilbert calculus. ∇

Intuitively, in a persistent Hilbert calculus, if a formula is Q-globally derivable from a set of closed formulas, then it is also locally derivable from the same set. Similarly, if a formula is O-globally derivable from a set of rigid formulas, then it is also locally derivable from the same set. That is, quantifier global rules do not bring anything new from a set of closed formulas and modal global rules do not bring anything new from a set of rigid formulas.

Lemma 6.6.4 *If a Hilbert calculus is Q-globally persistent, then it is vertically persistent. If a Hilbert calculus is O-locally persistent, then it is horizontally persistent.*

Proof. We prove only the first claim since the proof of the second is quite similar. Let H be a Hilbert calculus. Assume that

$$\Gamma^{\vdash^g_H}, \Psi \vdash^g_{\mathrm{Q}H} \varphi \lhd \pi \sqcap \mathrm{cfo}(\Psi)$$

and assume there is a corresponding derivation which contains N applications of rules in $R_{\mathrm{Q}g} \setminus R_\ell$; we show that the derivation can be transformed into a derivation of φ from $\Gamma^{\vdash^g_H} \cup \Psi$ with proviso $\mathrm{cfo}(\Psi)$ with $N-1$ applications of those rules.

Consider the first part

$$\langle \gamma_1, \pi_1 \rangle \ldots \langle \gamma_k, \pi_k \rangle$$

of the derivation of φ from $\Gamma^{\vdash^g_H} \cup \Psi$. Assume that γ_k is obtained by a rule r in $R_{\mathrm{Q}g} \setminus R_\ell$ and that no rules in $R_{\mathrm{Q}g} \setminus R_\ell$ were used before. Then, γ_k is $(op(\gamma_j))$ for some $j < k$. Consider the sequence

$$\langle \gamma_1, \pi_1 \rangle \langle (op(\gamma_1)), \pi_1 \rangle \ldots \langle \gamma_{k-1}, \pi_{k-1} \rangle \langle (op(\gamma_{k-1})), \pi_{k-1} \rangle \langle \gamma_k, \pi_k \rangle$$

where each π_i contains $\mathrm{cfo}(\Psi)$. We show that each pair in it can be derived from $\Gamma^{\vdash^g_H} \cup \Psi$. This will conclude the proof because γ_k is $(op(\gamma_j))$.

If $\gamma_i \in \Gamma^{\vdash^g_H}$, then $(op(\gamma_i))$ is also in $\Gamma^{\vdash^g_H}$. If $\gamma_i \in \Psi$, then $\Gamma^{\vdash^g_H}, \Psi \vdash^\ell (op(\gamma_i))$, using the third condition in Definition 6.6.1. Assume that γ_i is derived by means of an instance $\langle \{\psi'_1, \ldots, \psi'_n\}, \gamma_i, \pi'' \rangle$ of a rule in R_ℓ. Then, using fourth condition in Definition 6.6.1, we can conclude that

$$\{(op(\psi'_1)), \ldots, (op(\psi'_n))\} \vdash^\ell_H (op(\gamma_i)) \lhd \pi''$$

and so we have the result by induction. $\lhd$

Example 6.6.5 Consider the Hilbert calculus presented in Example 6.3.6. The Q-global rule

$$\langle \emptyset, (\xi \Rightarrow (\forall x\, \xi)), \{x \not\in \xi\} \rangle$$

ensures vertical persistency, and the O-global rule

$$\langle \emptyset, (\xi \Rightarrow (\Box \xi)), \mathrm{rig}(\xi) \rangle$$

guarantees horizontal persistency. ▽

A preservation result for persistence is now established. We show that persistence is preserved by fibring Q-globally persistent and O-globally persistent Hilbert calculi.

Theorem 6.6.6 *Let H' and H'' be persistent Hilbert calculi. If H' and H'' are Q-globally persistent and O-globally persistent then their fibring $H' \cup H''$ is also a persistent Hilbert calculus.*

Proof. By Theorem 6.6.2 and Lemma 6.6.4. ◁

We now consider Hilbert calculi with implication and Hilbert calculi with equivalence. This is just an adaptation of Definitions 2.3.7, 2.3.11 and 2.3.15.

Definition 6.6.7 A Hilbert calculus H is said to be a *Hilbert calculus with implication* if

- the signature contains a connective $\Rightarrow$ of arity 2;
- the *metatheorem of modus ponens* (*MTMP*) holds, that is, for every globally deductively closed Γ and $\{\varphi_1, \varphi_2\} \subseteq L(\Sigma)$

$$\frac{\Gamma \vdash_H^\ell (\varphi_1 \Rightarrow \varphi_2) \lhd \pi}{\Gamma, \varphi_1 \vdash_H^\ell \varphi_2 \lhd \pi} \qquad (\mathit{MTMP})$$

- the *metatheorem of deduction* (*MTD*) holds, that is, for every globally deductively closed Γ and $\{\varphi_1, \varphi_2\} \subseteq L(\Sigma)$

$$\frac{\Gamma, \varphi_1 \vdash_H^\ell \varphi_2 \lhd \pi}{\Gamma \vdash_H^\ell (\varphi_1 \Rightarrow \varphi_2) \lhd \pi} \qquad (\mathit{MTD})$$

A logic system $\mathcal{L}$ is said to be a *logic system with implication* whenever the underlying Hilbert calculus is with implication. ▽

The first-order version of Definition 2.3.17 is the following:

Definition 6.6.8 A Hilbert calculus H with implication $\Rightarrow$ is said to be a *Hilbert calculus with equivalence* if

- its signature contains a connective $\Leftrightarrow$ of arity 2;
- the two *metatheorems of biconditional*, (*MTB1*) and (*MTB2*), hold, that is, for every $\Gamma \cup \{\varphi_1, \varphi_2\} \subseteq L(\Sigma)$,

$$\frac{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2) \lhd \pi \qquad \Gamma \vdash^{\ell}_{H} (\varphi_2 \Rightarrow \varphi_1) \lhd \pi}{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Leftrightarrow \varphi_2) \lhd \pi} \qquad (MTB1)$$

$$\frac{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Leftrightarrow \varphi_2) \lhd \pi}{\Gamma \vdash^{\ell}_{H} (\varphi_1 \Rightarrow \varphi_2) \lhd \pi \qquad \Gamma \vdash^{\ell}_{H} (\varphi_2 \Rightarrow \varphi_1) \lhd \pi} \qquad (MTB2)$$

- the three *metatheorems of substitution of equivalents*, (*MTSE1*), (*MTSE2*) and (*MTSE3*) hold, that is, for every connective c, quantifier qx, and modality o of arity k and $\Gamma \cup \{\varphi_1, \varphi_2, \ldots, \varphi_k, \varphi'_1, \varphi'_2, \ldots, \varphi'_k\} \subseteq L(\Sigma)$,

$$\frac{\Gamma \vdash^{\ell}_{H} (\varphi_i \Leftrightarrow \varphi'_i) \lhd \pi\,, \text{ for } i = 1, \ldots, k}{\Gamma \vdash^{\ell}_{H} ((c(\varphi_1, \ldots, \varphi_k)) \Leftrightarrow (c(\varphi'_1, \ldots, \varphi'_k))) \lhd \pi} \qquad (MTSE1)$$

$$\frac{\Gamma^{\vdash^{g}_{\mathrm{Q}H}} \vdash^{\ell}_{H} (\varphi_i \Leftrightarrow \varphi'_i) \lhd \pi\,, \text{ for } i = 1, \ldots, k}{\Gamma^{\vdash^{g}_{\mathrm{Q}H}} \vdash^{\ell}_{H} (((qx)(\varphi_1, \ldots, \varphi_k)) \Leftrightarrow ((qx)(\varphi'_1, \ldots, \varphi'_k))) \lhd \pi} \qquad (MTSE2)$$

$$\frac{\Gamma^{\vdash^{g}_{\mathrm{O}H}} \vdash^{\ell}_{H} (\varphi_i \Leftrightarrow \varphi'_i) \lhd \pi\,, \text{ for } i = 1, \ldots, k}{\Gamma^{\vdash^{g}_{\mathrm{O}H}} \vdash^{\ell}_{H} ((o(\varphi_1, \ldots, \varphi_k)) \Leftrightarrow (o(\varphi'_1, \ldots, \varphi'_k))) \lhd \pi} \qquad (MTSE3)$$

A logic system $\mathcal{L}$ with implication is said to be a *logic system with equivalence* whenever the underlying Hilbert calculus is with equivalence. ▽

It is worth noting that the last item of the definition above is a generalization of Proposition 2.3.24.

Example 6.6.9 The Hilbert calculus presented in Example 6.3.6 is with implication and equivalence. ▽

The next theorem states that preservation of implication and equivalence by fibring holds, providing that the corresponding connectives are shared. This situation is analogous to the propositional based case.

Theorem 6.6.10 *The fibring of Hilbert calculi with implication is a Hilbert calculus with implication, provided that implication is shared by their signatures.*

The fibring of Hilbert calculi with equivalence is a Hilbert calculus with equivalence, provided that both implication and equivalence are shared by their signatures.

The distinction between quantifier and modal rules is also relevant in the notion of congruent Hilbert calculus introduced below.

Definition 6.6.11 A Hilbert calculus H is said to be *congruent* whenever the following conditions hold:

- for every Q-globally closed set $\Gamma' \subseteq L(\Sigma)$, O-globally closed set $\Gamma'' \subseteq L(\Sigma)$, formulas $\varphi_1, \varphi_1', \ldots, \varphi_k, \varphi_k'$ in $L(\Sigma)$, and $c \in C_k$,

$$\frac{\Gamma', \Gamma'', \varphi_i \vdash_H^\ell \varphi_i' \lhd \pi \qquad \Gamma', \Gamma'', \varphi_i' \vdash_H^\ell \varphi_i \lhd \pi \quad \text{ for } i = 1, \ldots, k}{\Gamma', \Gamma'', (c(\varphi_1, \ldots, \varphi_k)) \vdash_H^\ell (c(\varphi_1', \ldots, \varphi_k')) \lhd \pi}$$

- for every Q-globally closed set $\Gamma \subseteq L(\Sigma)$, formulas $\varphi_1, \varphi_1', \ldots, \varphi_k, \varphi_k'$ in $L(\Sigma)$, $qx \in Q_k$, $x \in X$,

$$\frac{\Gamma, \varphi_i \vdash_H^\ell \varphi_i' \lhd \pi \qquad \Gamma, \varphi_i' \vdash_H^\ell \varphi_i \lhd \pi \quad \text{ for } i = 1, \ldots, k}{\Gamma, ((qx)(\varphi_1, \ldots, \varphi_k)) \vdash_H^\ell ((qx)(\varphi_1', \ldots, \varphi_k')) \lhd \pi}$$

- for every O-globally closed set $\Gamma \subseteq L(\Sigma)$, formulas $\varphi_1, \varphi_1', \ldots, \varphi_k, \varphi_k'$ in $L(\Sigma)$, and $o \in O_k$,

$$\frac{\Gamma, \varphi_i \vdash_H^\ell \varphi_i' \lhd \pi \qquad \Gamma, \varphi_i' \vdash_H^\ell \varphi_i \lhd \pi \quad \text{ for } i = 1, \ldots, k}{\Gamma, (o(\varphi_1, \ldots, \varphi_k)) \vdash_H^\ell (o(\varphi_1', \ldots, \varphi_k')) \lhd \pi}$$

∇

A logic system $\mathcal{L}$ is said to be *congruent* whenever the underlying Hilbert calculus is congruent. ∇

It is not difficult to understand why the set is required to be Q-globally closed or O-globally closed. Note that in first-order logic, if $\Gamma = \{\varphi, \psi\}$ we have

$$\Gamma, \varphi \vdash^\ell \psi \text{ and } \Gamma, \psi \vdash^\ell \varphi$$

but in general we do not have

$$\Gamma, (\forall x\, \varphi) \vdash^\ell (\forall x\, \psi).$$

And, in modal logic, if $\Gamma = \{\varphi, \psi\}$ we have

$$\Gamma, \varphi \vdash^\ell \psi \text{ and } \Gamma, \psi \vdash^\ell \varphi$$

but in general we do not have

$$\Gamma, (\Box\, \varphi) \vdash^\ell (\Box\, \psi).$$

The following two results generalize Proposition 2.3.30 and Theorem 2.3.32.

Proposition 6.6.12 *Congruence holds in Hilbert calculi with equivalence.*

A preservation result for congruence is now established. We show that congruence is preserved by fibring Hilbert calculi with equivalence, whenever both implication and equivalence are shared.

Theorem 6.6.13 *Let H' and H'' be congruent Hilbert calculi. If H' and H'' are with equivalence and they share implication and equivalence then their fibring $H' \cup H''$ is also a congruent Hilbert calculus.*

Proof. By Proposition 6.6.12 and Proposition 6.6.10. $\lhd$

We now present the classes of Hilbert calculi for equality and inequality. Recall that the symbols $\approx$ and $\not\approx$ are assumed to be always available in every first-order based logic and that their semantics do not change from interpretation structure to interpretation structure (see Definition 6.2.1). However, up to now, we have not assumed anything about them from the deductive point of view.

Definition 6.6.14 A Hilbert calculus H is said to be *for equality* if, for every $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$, $t, t_1, t'_1, \ldots, t_k, t'_k \in T(\Sigma)$, $f \in F_k$ and $p \in P_k$:

- $\vdash^{\ell}_{H} (t \approx t)$;
- $(t_1 \approx t_2) \vdash^{\ell}_{H} (t_2 \approx t_1)$;
- $(t_1 \approx t_2), (t_2 \approx t_3) \vdash^{\ell}_{H} (t_1 \approx t_3)$;
- $$\frac{\Gamma \vdash^{\ell}_{H} (t_j \approx t'_j) \lhd \pi \quad \text{for } j = 1, \ldots, k}{\Gamma \vdash^{\ell}_{H} (f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k)) \lhd \pi}$$
- $$\frac{\Gamma \vdash^{\ell}_{H} (t_j \approx t'_j) \lhd \pi \quad \text{for } j = 1, \ldots, k}{\Gamma, p(t_1, \ldots, t_k) \vdash^{\ell}_{H} p(t'_1, \ldots, t'_k) \lhd \pi}$$
- $$\frac{\Gamma, (t \approx \mathbf{i}) \vdash^{\ell}_{H} \varphi \lhd \pi}{\Gamma \vdash^{\ell}_{H} \varphi \lhd \pi}$$

  where the individual symbol $\mathbf{i} \in \mathit{Ind}$ does not occur in the rules of H and $\pi(\rho) = 0$ whenever $\mathbf{i}$ occurs in $\rho(\Gamma)$ or in $\rho(\varphi)$.

The first three requirements impose that equality is an equivalence relation. The fourth and fifth requirements impose that equality is a congruence relation both for function and predicate symbols, respectively.

A logic system $\mathcal{L}$ is said to be *for equality* whenever the underlying Hilbert calculus is. $\triangledown$

Observe that, as a consequence of the first four conditions in Definition 6.6.14, equality is a congruence relation. The last condition expresses a well known derived rule in ordinary first-order logic with equality that is reasonable to assume of any first-order based logic for equality.

Definition 6.6.15 A Hilbert calculus H for equality is said to be *for inequality* if, for every $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$ and $t_1, t_2 \in T(\Sigma)$:

- $$\frac{\Gamma \vdash^{\ell}_{H} (t_1 \approx t_2) \lhd \pi \qquad \Gamma \vdash^{\ell}_{H} (t_1 \not\approx t_2) \lhd \pi}{\Gamma \vdash^{\ell}_{H} \varphi \lhd \pi}$$
- $$\frac{\Gamma, (t_1 \approx t_2) \vdash^{\ell}_{H} \varphi \lhd \pi \qquad \Gamma, (t_1 \not\approx t_2) \vdash^{\ell}_{H} \varphi \lhd \pi}{\Gamma \vdash^{\ell}_{H} \varphi \lhd \pi}$$

A logic system $\mathcal{L}$ for equality is said to be *for inequality* whenever the underlying Hilbert calculus is. ▽

The two conditions relate inequality with equality as expected when nothing is assumed about the available connectives. In classical first-order logic, the two requirements for inequality are equivalent to saying that

$$(t_1 \not\approx t_2) \quad \text{and} \quad (\neg(t_1 \approx t_2))$$

are (locally) interderivable for every terms t_1 and t_2.

Example 6.6.16 The Hilbert calculus presented in Example 6.3.6 is for equality and inequality. ▽

To establish sufficient conditions for the preservation of equality and inequality by fibring we have to consider some other properties of Hilbert calculi. We first consider Hilbert calculi with strong equality. The only difference between the requirements for equality and those for strong equality is that the last condition in Definition 6.6.14 is replaced with a new one. In the new requirement only rules (and not inferences) are involved.

Definition 6.6.17 A Hilbert calculus with implication is said to be with *strong equality* if the first four conditions in Definition 6.6.14 are fulfilled and, in addition,

$$\langle \{(\theta \approx x) \Rightarrow \xi\}, \xi, \pi \rangle \in R_g$$

where, for each signature Σ, $\pi_\Sigma(\rho) = 1$ if x does not occur in $\rho(\xi)$.

A logic system $\mathcal{L}$ with implication is said to with *strong equality* whenever the underlying Hilbert calculus is. ▽

Lemma 6.6.18 *In any Hilbert calculus, congruence for function symbols holds if and only if*

$$(t_1 \approx t'_1), \dots, (t_k \approx t'_k) \vdash^{\ell}_{H} (f(t_1, \dots, t_k) \approx f(t'_1, \dots, t'_k))$$

for every $f \in F_k$, $k > 0$.

In any Hilbert calculus with equivalence, congruence for predicate symbols holds if and only if

$$(t_1 \approx t'_1), \ldots, (t_k \approx t'_k) \vdash^{\ell}_{H} (p(t_1, \ldots, t_k) \Leftrightarrow p(t'_1, \ldots, t'_k))$$

for every $p \in P_k$, $k > 0$.

Proof. Congruence for function symbols implies

$$(t_1 \approx t'_1), \ldots, (t_k \approx t'_k) \vdash^{\ell}_{H} (f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k))$$

for $\Gamma = \{(t_1 \approx t'_i) : i = 1, \ldots, k\}$ and $\pi = \mathbf{1}$.

Assume now $(t_1 \approx t'_1), \ldots, (t_k \approx t'_k) \vdash^{\ell}_{H} (f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k))$ and $\Gamma \vdash^{\ell}_{H} (t_i \approx t'_i) \lhd \pi$, for $i = 1, \ldots, k$. Then, we can clearly construct a proof of

$$(f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k))$$

from Γ by suitably "putting together" the proofs of $(t_i \approx t'_i)$ from Γ and the proof of $(f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k))$ from the equalities $(t_i \approx t'_i)$. Since each of the former proofs is constrained by π and the latter is constrained by $\mathbf{1}$, we eventually obtain $\Gamma \vdash^{\ell}_{H} (f(t_1, \ldots, t_k) \approx f(t'_1, \ldots, t'_k)) \lhd \pi$.

By the properties of implication and of equivalence, congruence for predicate symbols is equivalent to

$$\frac{\Gamma \vdash^{\ell}_{H} (t_i \approx t'_i) \lhd \pi \text{ for } i = 1, \ldots, k}{\Gamma \vdash^{\ell}_{H} p(t_1, \ldots, t_k) \Leftrightarrow p(t'_1, \ldots, t'_k) \lhd \pi}$$

and hence the proof that congruence for predicate symbols is equivalent to

$$(t_1 \approx t'_1), \ldots, (t_k \approx t'_k) \vdash^{\ell}_{H} (p(t_1, \ldots, t_k) \Leftrightarrow p(t'_1, \ldots, t'_k))$$

is quite similar to that above. ◁

A preservation result for strong equality follows. We show that strong equality is preserved by fibring Hilbert calculi with equivalence, provided that both implication and equivalence are shared.

Theorem 6.6.19 *Let H' and H'' be Hilbert calculi with strong equality. If H' and H'' are with equivalence and they share implication and equivalence then their fibring $H' \cup H''$ is also a Hilbert calculus with strong equality.*

Proof. Use Proposition 6.5.5 in order to have that the first three conditions hold in the fibring. The same proposition and Lemma 6.6.18 can be used in order to have that the last condition is preserved. The fibring includes $\langle \{(\theta \approx x) \Rightarrow \xi\}, \xi, \pi \rangle$ as a global rule simply because of Definition 6.5.1. ◁

Next, we relate the properties of strong equality and uniformity with equality.

Proposition 6.6.20 *Every uniform Hilbert calculus with strong equality is a Hilbert calculus with equality.*

Proof. Let H be uniform Hilbert calculus with strong equality. From the definition of strong equality, H is with implication.

Assume

$$\Gamma, (t \approx \mathbf{i}) \vdash^{\ell}_{H} \varphi \lhd \pi$$

where the invariant $\mathbf{i}$ does not occur in the rules of H and $\pi(\rho) = 0$ whenever $\mathbf{i}$ occurs in $\rho(\Gamma)$ or in $\rho(\varphi)$. By compactness we also have

$$\{\gamma_1, \ldots, \gamma_k\}, (t \approx \mathbf{i}) \vdash^{\ell}_{H} \varphi \lhd \pi$$

for some $\{\gamma_1, \ldots, \gamma_k\} \subseteq \Gamma$. Moreover, since we are assuming that H is with implication, we also have

$$\vdash^{\ell}_{H} ((t \approx \mathbf{i}) \Rightarrow \varphi^*) \lhd \pi, \quad \text{for } \varphi^* = (\gamma_1 \Rightarrow (\gamma_2 \Rightarrow \ldots \Rightarrow (\gamma_k \Rightarrow \varphi)\ldots))$$

Since we are considering a uniform Hilbert calculus, Proposition 6.3.14 implies

$$\vdash^{\ell}_{H} ((t \approx x) \Rightarrow \varphi^*) \lhd \pi$$

which implies in turn

$$\vdash^{g}_{H} ((t \approx x) \Rightarrow \varphi^*) \lhd \pi \qquad (*).$$

Consider now the rule for strong equality in Definition 6.6.17, and write π_0 for the proviso in it. Given any substitution σ such that $\sigma(\theta) = t$ and $\sigma(\xi) = \varphi^*$, $(*)$ implies

$$\vdash^{g}_{H} \varphi^* \lhd \pi \sqcap \pi_0\sigma \;\text{ and }\; \vdash^{\ell}_{H} \varphi^* \lhd \pi \sqcap \pi_0\sigma.$$

Given any ground substitution ρ over Σ, we have $\pi_0\sigma(\rho) = \pi_0(\hat{\rho} \circ \sigma) = 1$ if and only if $\mathbf{i}$ does not occur in $\rho(\sigma(\xi))$, that is, if and only if $\mathbf{i}$ does not occur in $\rho(\varphi^*)$. Thus, since $\pi(\rho) = 0$ whenever i occurs in $\rho(\varphi^*)$, we have that $\pi = \pi \sqcap \pi_0\sigma$ and hence $\vdash^{\ell}_{h} \varphi^* \lhd \pi$.

Using the properties of implication again, we have $\{\gamma_1, \ldots, \gamma_k\} \vdash^{\ell}_{H} \varphi \lhd \pi$ and $\Gamma \vdash^{\ell}_{H} \varphi \lhd \pi$. ◁

We now establish the preservation of uniformity by fibring. The preservation of this property is used to prove the preservation of equality.

Theorem 6.6.21 *The fibring of two uniform Hilbert calculi is also a uniform Hilbert calculus.*

Proof. Straightforward from Definition 6.3.13. ◁

We can now establish a sufficient condition for the preservation of equality by fibring, as well as a sufficient condition for the preservation of inequality.

Theorem 6.6.22 *Let H' and H'' be Hilbert calculi with equality. If they are uniform, with strong equality and equivalence then their fibring $H' \cup H''$ is also with equality.*

Proof. By Theorems 6.6.19, 6.6.20 and 6.6.21. ◁

Theorem 6.6.23 *Let H' and H'' be Hilbert calculi with inequality. If they are with equality, implication and share implication then their fibring $H' \cup H''$ is also with inequality.*

Proof. It is straightforward to verify that the clauses in Definition 6.6.15, respectively, hold if and only if:

- $(t \approx t'), (t \not\approx t') \vdash^{\ell}_{H} \varphi$;
- $\vdash^{\ell}_{H} (((t \approx t') \Rightarrow \varphi) \Rightarrow (((t \not\approx t') \Rightarrow \varphi) \Rightarrow \varphi))$.

Again, using Proposition 6.5.5, we obtain the intended preservation. ◁

Observe that the preservation results stated in the above propositions only refer Hilbert calculi. Clearly, using these propositions, it is straightforward to establish similar preservation results for logic systems.

We end this section with the proof of the preservation of fullness by fibring. Recall this is a property of logic systems (see Definition 6.4.8). This result is the first-order counterpart of Theorem 3.3.16.

Theorem 6.6.24 *The fibring of two full logic systems is also a full logic system.*

Proof. Let $\mathcal{L}'$ and $\mathcal{L}''$ be two full logic systems. We have to show that every interpretation structure s over $\Sigma' \cup \Sigma''$ appropriate for $H(\Sigma' \cup \Sigma'')$ is in $A(M)$. That is, we have to show that

$$s|_{\Sigma'} \in A'(M') \text{ and } s|_{\Sigma''} \in A''(M'').$$

Indeed, s is appropriate for both $H'(\Sigma')$ and $H''(\Sigma'')$, and, hence, $s|_{\Sigma'}$ is appropriate for $H'(\Sigma')$ and $s|_{\Sigma''}$ is appropriate for $H''(\Sigma'')$. Given the fullness of $\mathcal{L}'$ and $\mathcal{L}''$, $s|_{\Sigma'} \in A'(M')$ and $s|_{\Sigma''} \in A''(M'')$. ◁

6.6.2 Completeness

In this section, we concentrate on the problem of preservation of completeness by fibring. Recall that we are considering sound logic systems and that soundness is preserved by fibring by the very definition of fibring. We prove that completeness is indeed preserved by fibring, under some natural assumptions that are fulfilled in a wide class of logics encompassing the most common first-order based logics.

To prove the preservation of completeness by fibring, as we did for the propositional based case, we first state a completeness theorem, Theorem 6.6.25, which establishes sufficient conditions for a logic system to be complete. However, in general, the properties of logics systems considered in this theorem are not always preserved by fibring. Hence, we then characterize a class of logic systems which is closed under fibring and such that every element of it enjoys the sufficient conditions for completeness referred in Theorem 6.6.25. The preservation completeness theorem, Theorem 6.6.27, then states that completeness is preserved by fibring logics from that class.

Recall that completeness is stated only for ground formulas since there is no semantic counterpart to provisos. Hence, provisos will not appear. Observe that for any set Γ of ground formulas and ground formula φ, if, for instance, $\Gamma \vdash_H^\ell \varphi \lhd \pi$, then

$$\Gamma \vdash_H^\ell \varphi \lhd \mathbf{1}_\Sigma.$$

We now state the completeness theorem referred above.

Theorem 6.6.25 *Every full, congruent, persistent, and uniform logic system with equality and inequality is complete.*

It is worthwhile to compare this result with Theorem 3.3.15 for propositional based logics to see what are the additional requirements for the first-order case.

Herein, we only sketch the proof of Theorem 6.6.25. For more details see [242]. The proof uses the Henkin construction (Leon Henkin introduced the technique, see [144, 145]) briefly described in the sequence.

We assume as given once and for all the sound logic system

$$\mathcal{L} = \langle \Sigma, M, A, R_\ell, R_{\mathrm{Q}g}, R_{\mathrm{O}g}, R_g \rangle$$

with underlying Hilbert calculus H, which is assumed to be congruent, persistent, uniform and for equality and inequality, and where $\Sigma = \langle \mathit{Ind}, F, P, C, Q, O \rangle$.

Before describing the Henkin construction we need to introduce the following notions. We say that $\Gamma \subseteq L(\Sigma)$ is *φ-consistent*, $\varphi \in gL(\Sigma)$, whenever $\Gamma \not\vdash_H^\ell \varphi$ and we say that it is *φ-maximal consistent* (φ-m.c.s.) if it is φ-consistent and no proper extension of it is φ-consistent. Moreover, Γ is said to be *consistent* or *maximal consistent* if, respectively, there is a φ such that Γ is φ-consistent, or there is a φ such that Γ is φ-maximal consistent. Any φ-m.c.s. including Γ is said to be a *φ-maximal consistent extension* (φ-m.c.e.) of Γ. Every φ-m.c.e. of Γ is said to be an m.c.e. of Γ.

Given any set E such that $F_0 \cap E = \mathit{Ind} \cap E = \emptyset$, we denote by

$$\Sigma_E \text{ and } H_E$$

the signature obtained from Σ by replacing Ind by $\mathit{Ind} \cup E$ and the Hilbert calculus obtained from H where each rule is seen as a rule over Σ_E, respectively. A set $\Gamma \subseteq gL(\Sigma_E)$ is said to be an *E-Henkin set* whenever:

- Γ is an m.c.s. in $L(\Sigma_E)$;
- for every term $t \in gT(\Sigma)$, there is a $d \in E$ such that $(t \approx d) \in \Gamma$;
- for every $d \in E$, there is a term $t \in gT(\Sigma)$ such that $(t \approx d) \in \Gamma$;
- $\{(d \not\approx d') \ : \ d, d' \in E\} \subseteq \Gamma$.

Γ is said to be an *E-pre-Henkin set* if the two last conditions above are fulfilled.

Given $\Gamma \subseteq gL(\Sigma_E)$, the *Q-kernel* and of Γ, written $K_Q(\Gamma)$, is defined by

$$\begin{array}{rcl} K_Q(\Gamma) & = & \{\varphi \in \Gamma : \varphi \text{ is a first-order formula and } \varphi \in cL(\Sigma)\} \cup \\ & & \{(t \approx d) \in \Gamma : t \in cT(\Sigma) \text{ and } d \in E\} \cup \\ & & \{(d \not\approx d') : \text{ distinct } d, d' \in E \text{ and} \\ & & \ \text{there exists } t, t' \in cT(\Sigma) \text{ such that } (t \approx d) \in \Gamma \text{ and } (t' \approx d') \in \Gamma\}\,. \end{array}$$

We now begin to describe the Henkin construction. Let $\Gamma_0 \subseteq gL(\Sigma)$ be a consistent and globally closed set and let D be a set with cardinality greater that of $gT(\Sigma)$ and such that

- $F_0 \cap D = \emptyset$
- $Ind \cap D = \emptyset$.

We define an appropriate interpretation structure

$$s = \langle U, \mathbb{V}, W, \alpha, \omega, D, \mathcal{E}, \mathcal{B}, [\cdot]\rangle$$

for the Hilbert calculus H as follows:

- $U = \{u \subseteq gL(\Sigma_D) : u \text{ is a } E\text{-Henkin set for some } E \subseteq D \text{ and } \Gamma_0 \subseteq u\}$;
- $\mathbb{V} = \{\vartheta_u : u \in U\}$
  where $\vartheta_u \in E^{X \cup Ind}$ such that $\vartheta_u(t) = t \in u$ for every $t \in X \cup Ind$;
- $W = \{K_Q(u) : u \in U\}$;
- $\omega(u) = K_Q(u)$;
- $\alpha(u) = \vartheta_u$;
- $\mathcal{E} = \{|t| : t \in gT(\Sigma)\}$
  where $|t| : U \to D$ is a map such that $|t|(u) = d$ whenever $(t \approx d) \in u$;
- $\mathcal{B} = \{|\varphi| : \varphi \in gL(\Sigma)\}$
  where $|\varphi| : U \to \{0, 1\}$ is a map such that $|\varphi|(u) = 1$ if $\varphi \in u$;

- the interpretation map $[\cdot]$ is such that
  - for $x \in X$, $i \in I$, and $\vartheta \in \mathbb{V}$,
  $$[x]_\vartheta = \vartheta(x)\,, \qquad [i]_\vartheta = \vartheta(i)\,;$$
  - for $f \in F_k$, $k \geq 0$, and $u \in U_w$,
  $$[f]_w(|t_1|(u), \ldots, |t_k|(u)) = |f(t_1, \ldots, t_k)|(u);$$
  - for every $u \in U$,
  $$[\approx](|t_1|(u), |t_2|(u)) = 1 \text{ if } |t_1|(u) = |t_2|(u)\,;$$
  $$[\not\approx](|t_1|(u), |t_2|(u)) = 1 \text{ if } |t_1|(u) \neq |t_2|(u)\,;$$
  - for $p \in P_k$, $k \geq 0$, and $u \in U_w$,
  $$[p]_w(|t_1|(u), \ldots, |t_k|(u)) = |p(t_1, \ldots, t_k)|(u)\,;$$
  - for every $c \in C_k$, $k \geq 0$, $w \in W$, and any assignment ϑ,
  $$[c]_{w\vartheta}(|\gamma_1|_{w\vartheta}, \ldots, |\gamma_k|_{w\vartheta}) = |c(\gamma_1, \ldots, \gamma_k)|_{w\vartheta};$$
  - for every $q \in Q_k$, $x \in X$, and $w \in W$,
  $$[qx]_w(|\gamma_1|_w, \ldots, |\gamma_k|_w) = |(qx)(\gamma_1, \ldots, \gamma_k)|_w;$$
  - for every $o \in O_k$ and any assignment ϑ,
  $$[o]_\vartheta(|\gamma_1|_\vartheta, \ldots, |\gamma_k|_\vartheta) = |o(\gamma_1, \ldots, \gamma_k)|_\vartheta;$$

  where $|\varphi|_w = |\varphi| \cap U_w$, $|\varphi|_\vartheta = |\varphi| \cap U_\vartheta$ and $|\varphi|_{w\vartheta} = |\varphi|_w \cap |\varphi|_\vartheta$, for every $\varphi \in gL(\Sigma)$, $w \in W$ and $\vartheta \in \mathbb{V}$.

In the interpretation structure s defined above we have that $|t| = [\![t]\!]_s$ for each $t \in gT(\Sigma)$, and $|\varphi| = [\![\varphi]\!]_s$ for every $\varphi \in gL(\Sigma)$. Furthermore,

$$|\varphi| = U \text{ if and only if } \varphi \in \Gamma_0.$$

The proof of the fact that s is indeed appropriate for the Hilbert calculus H follows from the assertions above using also the following result:

Lemma 6.6.26 *Let $E \subseteq D$. Then any E-pre-Henkin set Γ φ-consistent in H_E can be extended to a E^*-Henkin set*

$$\Gamma^* \subseteq gL(\Sigma_{E^*})$$

where $E \subseteq E^* \subseteq D$ and Γ^* is φ-m.c.e. of Γ in H_{E^*}.

Using the above results Theorem 6.6.25 is established proving that:

- if $\Gamma \not\vdash^{\ell}_{H} \varphi$ then there is an appropriated structure s for H and $u \in U$ such that $u \in [\![\gamma]\!]_s$ for all $\gamma \in \Gamma$ and $u \notin [\![\varphi]\!]_s$;
- if $\Gamma \not\vdash^{g}_{H} \varphi$ then there is an appropriated structure s for H such that $[\![\gamma]\!]_s = U$ for all $\gamma \in \Gamma$, but $[\![\varphi]\!]_s \subseteq U \setminus \{u\}$.

We now state the completeness preservation theorem.

Theorem 6.6.27 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be two complete logic systems. If $\mathcal{L}'$ and $\mathcal{L}''$ are full, uniform, Q-globally and O-globally persistent, with implication, equivalence, strong equality and inequality, and such that the implication and equivalence are shared, then their fibring $\mathcal{L}' \cup \mathcal{L}''$ is a complete logic system.*

Proof. From Theorem 6.6.24, $\mathcal{L}' \cup \mathcal{L}''$ is full. From Theorem 6.6.21, $\mathcal{L}' \cup \mathcal{L}''$ is uniform. From Theorem 6.6.2 and Lemma 6.6.4, $\mathcal{L}' \cup \mathcal{L}''$ is persistent. From Theorem 6.6.22, $\mathcal{L}' \cup \mathcal{L}''$ is with equality. From Theorem 6.6.23, $\mathcal{L}' \cup \mathcal{L}''$ is with inequality. From Theorem 6.6.10, $\mathcal{L}' \cup \mathcal{L}''$ is with equivalence. Therefore, from Theorem 6.6.12, $\mathcal{L}' \cup \mathcal{L}''$ is congruent.

Hence, $\mathcal{L}' \cup \mathcal{L}''$ is full, congruent, persistent, uniform and with equality and inequality. Using Theorem 6.6.25, we conclude that $\mathcal{L}' \cup \mathcal{L}''$ is complete. $\triangleleft$

6.7 Final remarks

In this chapter we addressed the problem of fibring first-order based logics. Once again we adopted an homogeneous scenario. Both original logics are endowed with Hilbert calculi and the same kind of semantics.

Several problems have to be dealt with when extending fibring to the first-order setting. The interaction between schema variables substitution and quantifiers, for instance, can have undesirable consequences at the deductive level. To cope with this problem we introduced the notion of inference rule with proviso. The proviso ensures that substitutions are safely handled when the rule is used in a derivation.

The semantic structures are powerset algebras both for the modal operators and the quantifiers. We took advantage of the fact that (general) Kripke structures induce powerset algebras and that quantifiers can be seen as modalities. This option means that the semantic structures in this chapter are not as abstract as the ones in Chapter 3.

An essential step toward a more abstract semantics will be to understand fully what is the fibring of cylindrical algebras for first-order logics (see [146]). In order to deal with the general case we need a general notion of cylindrical algebra for coping with more general quantifiers.

In this chapter, we do not use categories. The reason is to present the specific issues of first-order based logics in a known context. Moreover, we believe that the categorial effort is only worthwhile when the semantic structures are more algebraic in nature (not only powerset algebras).

We were able to establish sufficient conditions for preservation of completeness by starting to prove a general theorem of completeness. Along the way we also proved the preservation of some metatheorems such as the metatheorem of deduction and the metatheorem of substitution of equivalents. Once more we did not consider logics that are not truth-functional.

Instead of using Hilbert calculi we could have used other kinds of deductive systems. For example, in [225], fibring of first-order based logics is investigated adopting natural deductive systems.

We believe that more preservation properties can be investigated. The results on preservation of interpolation seems to be easily extended to this context.

Chapter 7

Fibring higher-order logics

The applications that motivate our approach to combination of logics require sometimes the use of arbitrary higher-order quantifiers, as well as arbitrary modal-like operators. In this chapter, the concept of fibring is extended to higher-order logics endowed with arbitrary modalities and binding operators. The approach used herein is more general than the one introduced in Chapter 6. The generality comes from the categorical semantics where the collection of truth-values is a topos. The transference results are also simpler than in the first-order case. However, one has to pay the price in the sense that this chapter is not easily read without some knowledge of category theory.

This chapter defines a wide class of logic systems equipped with topos semantics and with Hilbert calculi. Since the use of topos theory is fundamental herein, the reader should be acquainted with topos theory and local set theory. Moreover, it is more convenient to present the corresponding notions of fibring as categorical constructions. We refer the reader to [134] for topos theory with a logic flavor and [187] for category theory and topos theory. We also use [18] for local set theory and relationship with higher-order logics. Finally, we refer to [15] for the notion of cocartesian lifting.

It is worth noting that the class of logics to be studied here encompasses many commonly used logics, such as propositional logics, modal logics, quantification logics, typed lambda calculi and higher-order logics. Arbitrary modalities and binding operators are allowed, as well as any choice of rigid (world-independent) as well as flexible (world-dependent) function symbols.

The deduction mechanism considered herein, as in other chapters, is the Hilbert calculus style, but allowing in this case rules with provisos.

In what concerns semantics, the structures considered in this chapter generalize the usual topos semantics of higher-order logic. This generalization, while preserving the simplicity and elegance of the traditional topos semantics, is able to deal with arbitrary modalities, quantifiers and other binding operators. As done in other chapters, two entailment relations are defined: the local entailment as usually considered in categorical logic, and the global entailment necessary to deal with necessitation and generalization. Examples are given of familiar logics for which it is possible to lift the original semantics to the topos semantics level, while preserving the denotation of terms (and formulas). Thus, there is no loss of generality by assuming that the logics under consideration are endowed with the kind of topos semantics proposed here.

With respect to soundness, the novelty here is that the usual notion of soundness must be modified in the present framework. This is a consequence of the possibility of having empty domains interpreting the types (a basic feature of categorical semantics). It is proved that the basic example of *HOL* (a Hilbert-style axiomatization of intuitionistic higher-order logic) is sound with respect to a slightly generalized notion of topos semantics.

We establish a general completeness theorem about full logic systems: every full logic system with Hilbert calculus, including *HOL*, and enjoying the metatheorem of deduction, is complete. To prove this result we show that every consistent Hilbert calculus that includes *HOL* and enjoys the metatheorem of deduction has a canonical model. The construction of the canonical model is done as usual in categorical logic (see for instance [18]), but with the adaptations made necessary in view of the richer language we work with (arbitrary modalities and binding operators), and also in view of the two notions of entailment. This theorem plays an important role in the proof of the preservation of completeness by fibring. We first prove that the fibring of full logic systems endowed with Hilbert calculi that include *HOL* and with the metatheorem of deduction is also complete. Additionally, we show that, under some natural conditions, a full logic system is complete if and only if it can be conservatively enriched with *HOL*. Finally, as a consequence of this result, a second completeness preservation result is obtained: the fibring of two full, complete logic systems is also complete provided that conservativeness of *HOL*-enrichment is preserved. It is an open problem to find sufficient conditions for the preservation of *HOL*-enrichment.

This chapter is structured as follows. In Section 7.1 we introduce the relevant signatures. Section 7.2 presents the Hilbert calculus. Section 7.3 is dedicated to setting up the semantic notions. Section 7.4 introduces the notion of logic system, and we briefly discuss the related notions of soundness and completeness. In Section 7.5, a general completeness theorem is established. In Section 7.6, the notions of constrained and unconstrained fibring of logic systems are given, and it is shown that soundness is preserved by fibring and a completeness preservation result is obtained. Finally, in Section 7.7, we briefly discuss the main results described in the chapter, as well as some open problems related to the completeness preservation by fibring.

The contents of this chapter is based on [62].

7.1 Higher-order signatures

In order to cope with higher-order features, the notion of signature considered in this chapter has to be a bit more sophisticated than the ones considered in most of the previous chapters.

The notion of logic system associated to these higher-order signatures is carefully chosen having two main objectives in sight: (i) It should be sufficiently general as to include commonly used logics such as logics with arbitrary modalities and logics with first-order or even higher-order binding operators, of which quantifiers are but a special case; (ii) starting from any given logic as in (i), it should be possible to rephrase it in our settings while preserving the interpretation of terms and formulas at each model.

Goal (i) is motivated by the homogeneous scenario that we want to set up for fibring. Indeed, as it was done before, when combining logics one assumes that all of them are presented in the same style (with signatures of the same form, with models of the same kind and with deduction systems of the same nature). That is, in the homogeneous scenario, when combining two logics we assume that they are objects in the chosen category of logics. Therefore, the first sections of this chapter are dedicated to setting up the category **HLog** of logic systems where in Section 7.6 fibring is to be defined as a universal construction.

On the other hand, goal (ii) above addresses a preparation step: before combining two logics we have to present them as logic systems in the same category **HLog** so as to guarantee that this conversion step preserves all entailments of the given logic. Moreover, we would like to encode each model of the original logic in the obtained logic system. The same requirement applies to the language, in the following sense: each symbol of the original logic should be recognized in the obtained logic system, while preserving the language. At the deduction level, we assume that the logics are presented as Hilbert axiomatic systems, so the conversion of the deduction systems is automatic.

From now on, and until the end of this section, we concentrate on describing the formal languages we deal with.

Assume given once and for all the set $\mathbf{S}$ with distinguished elements $\mathbf{1}$ and $\mathbf{\Omega}$. The set $\Theta(\mathbf{S})$ is recursively defined as the minimum set satisfying the following:

1. $\mathbf{S} \subseteq \Theta(\mathbf{S})$;
2. if $n \geq 2$ and $\theta_1, \ldots, \theta_n \in \Theta(\mathbf{S})$ then $(\theta_1 \times \cdots \times \theta_n) \in \Theta(\mathbf{S})$;
3. if $\theta, \theta' \in \Theta(\mathbf{S})$ then $(\theta \to \theta') \in \Theta(\mathbf{S})$.

Technically, $\Theta(\mathbf{S})$ is a free algebra which is freely generated by $\mathbf{S}$, with a n-ary operation $\times$ (one operation for each $n \geq 2$) and with a binary operator $\to$ (recall Section 3.1 of Chapter 3). The expression θ^n will denote the n-th power of θ (the product of θ with itself n times) and by convention θ^0 is $\mathbf{1}$ and θ^1 is θ. The elements of $\mathbf{S}$ are called *base sorts* or *base types*. The elements of $\Theta(\mathbf{S})$ are called *sorts* or *types* over $\mathbf{S}$. Base types $\mathbf{1}$ and $\boldsymbol{\Omega}$ are called the *unit sort* and the *truth value sort*, respectively. Assume also as given once and for all the families:

- $\Xi = \{\Xi_\theta\}_{\theta \in \Theta(\mathbf{S})}$ where each Ξ_θ is a denumerable set;
- $X = \{X_\theta\}_{\theta \in \Theta(\mathbf{S})}$ where each X_θ is a denumerable set.

The elements of each Ξ_θ and X_θ are called, respectively, *schema variables* and *variables* of type θ. Assume that the sets $\Xi_\theta \cap \Xi_{\theta'}$ and $X_\theta \cap X_{\theta'}$ are empty, for all $\theta, \theta' \in \Theta(\mathbf{S})$ such that $\theta \neq \theta'$. Moreover, $\Xi_\theta \cap X_{\theta'}$ is empty for all $\theta, \theta' \in \Theta(\mathbf{S})$.

Definition 7.1.1 A *higher-order signature* is a triple $\Sigma = \langle \mathsf{R}, \mathsf{F}, \mathsf{Q} \rangle$ such that:

- $\mathsf{R} = \{\mathsf{R}_{\theta\theta'}\}_{\theta,\theta' \in \Theta(\mathbf{S})}$ where each $\mathsf{R}_{\theta\theta'}$ is a set;
- $\mathsf{F} = \{\mathsf{F}_{\theta\theta'}\}_{\theta,\theta' \in \Theta(\mathbf{S})}$ where each $\mathsf{F}_{\theta\theta'}$ is a set;
- $\mathsf{Q} = \{\mathsf{Q}_{\theta\theta'\theta''}\}_{\theta,\theta',\theta'' \in \Theta(\mathbf{S})}$ where each $\mathsf{Q}_{\theta\theta'\theta''}$ is a set. ∇

The elements of each $\mathsf{R}_{\theta\theta'}$ are called *rigid function symbols* of type $\theta\theta'$. The elements of each $\mathsf{F}_{\theta\theta'}$ are called *flexible function symbols* of type $\theta\theta'$. The elements of each $\mathsf{Q}_{\theta\theta'\theta''}$ are called (binding) *operator symbols* of type $\theta\theta'\theta''$. Typical examples of binding operations are quantification, lambda-abstraction and set comprehension. We assume that the members of the families constituting a signature are pairwise disjoint.

Since in this chapter we only deal with higher-order signatures, for simplicity, from now on "signature" will stand for "higher-order signature".

Definition 7.1.2 The family $T(\Sigma) = \{T(\Sigma)_\theta\}_{\theta \in \Theta(\mathbf{S})}$ is inductively defined as follows:

- $\Xi_\theta \cup X_\theta \subseteq T(\Sigma)_\theta$;
- if $\xi \in \Xi_\theta$, $x \in X_{\theta'}$ and $\xi' \in \Xi_{\theta'}$ then $\xi^x_{\xi'} \in T(\Sigma)_\theta$;
- if $r \in \mathsf{R}_{\theta\theta'}$ and $t \in T(\Sigma)_\theta$ then $(r\,t) \in T(\Sigma)_{\theta'}$;
- if $f \in \mathsf{F}_{\theta\theta'}$ and $t \in T(\Sigma)_\theta$ then $(f\,t) \in T(\Sigma)_{\theta'}$;
- if $q \in \mathsf{Q}_{\theta\theta'\theta''}$, $x \in X_\theta$ and $t \in T(\Sigma)_{\theta'}$ then $(q x\,t) \in T(\Sigma)_{\theta''}$;
- if $n \neq 1$ and $t_i \in T(\Sigma)_{\theta_i}$ for $i = 1, \ldots, n$ then $\langle t_1, \ldots, t_n \rangle \in T(\Sigma)_{\theta_1 \times \cdots \times \theta_n}$; in particular, $\langle\rangle \in T(\Sigma)_{\mathbf{1}}$;
- if $n \geq 2$ and $t \in T(\Sigma)_{\theta_1 \times \cdots \times \theta_n}$ then $(t)_i \in T(\Sigma)_{\theta_i}$ for $1 \leq i \leq n$. ∇

The elements of each $T(\Sigma)_\theta$ are called *terms* of type θ. Terms of type $\mathbf{\Omega}$ are also known as *formulas*. Terms without schema variables are called *ground terms*: $gT(\Sigma)_\theta$ denotes the set of ground terms of type θ. Formulas without schema variables are called *ground formulas*. We write $L(\Sigma)$ and $gL(\Sigma)$ for $T(\Sigma)_{\mathbf{\Omega}}$ and $gT(\Sigma)_{\mathbf{\Omega}}$, respectively.

We assume here the usual notions associated to binding operators. For instance, an occurrence of a variable x in a term t is said to be *bound* if and only if either it appears within the scope of some (binding) operator q applied to x, or it appears in a term of the form $\xi^x_{\xi'}$. Any other occurrence of x in t is *free*. Terms without free occurrences of variables are said to be *closed*. In particular, a formula without free occurrences of variables is said to be a *closed formula*.

The following examples show that the proposed notion of signature is rich enough to encompass a wide variety of logics. Moreover, the signatures (and, consequently, the generated languages) are not changed in any significant way.

Example 7.1.3 We consider modal propositional logic again. Fix a set $\mathbb{P}$ of propositional variables. Then:

- The members of the families R and F are empty, except:
  - $\mathsf{R}_{\mathbf{1}\mathbf{\Omega}} = \{\mathbf{f}, \mathbf{t}\}$;
  - $\mathsf{R}_{\mathbf{\Omega}\mathbf{\Omega}} = \{\neg\}$;
  - $\mathsf{R}_{\mathbf{\Omega}^2\mathbf{\Omega}} = \{\wedge, \vee, \Rightarrow\}$;
  - $\mathsf{F}_{\mathbf{1}\mathbf{\Omega}} = \mathbb{P}$;
  - $\mathsf{F}_{\mathbf{\Omega}\mathbf{\Omega}} = \{\Diamond, \Box\}$.
- All members of the family Q are empty. $\triangledown$

Example 7.1.4 We now consider propositional logic. As in Example 7.1.3, except $\mathsf{F}_{\mathbf{\Omega}\mathbf{\Omega}} = \emptyset$. Notice that we define the propositional symbols as flexible; this is justified for the purpose of fibring as will be explained in Section 7.6. $\triangledown$

Example 7.1.5 Given a first-order signature $\langle \mathcal{G}, \mathcal{P} \rangle$ where $\mathcal{G} = \{\mathcal{G}_n\}_{n\in\mathbb{N}}$ and $\mathcal{P} = \{\mathcal{P}_n\}_{n\in\mathbb{N}^+}$ (the families of sets of first-order function symbols and predicate symbols, respectively, of different arities) and a base sort $\mathbf{i}$ different from $\mathbf{1}$ and $\mathbf{\Omega}$:

- All members of the families R and F are empty, except:
  - $\mathsf{R}_{\mathbf{i}^n\mathbf{i}} = \mathcal{G}_n$ for $n \in \mathbb{N}$;
  - $\mathsf{R}_{\mathbf{\Omega}\mathbf{\Omega}} = \{\neg\}$;
  - $\mathsf{R}_{\mathbf{\Omega}^2\mathbf{\Omega}} = \{\wedge, \vee, \Rightarrow\}$;
  - $\mathsf{F}_{\mathbf{i}^n\mathbf{\Omega}} = \mathcal{P}_n$ for $n \in \mathbb{N}^+$.
- All members of the family Q are empty, except:

  - $\mathsf{Q}_{\mathbf{i}\boldsymbol{\Omega}\boldsymbol{\Omega}} = \{\exists, \forall\}$.

Again, since we intend to combine logics including modalities, functions are defined to be rigid and predicates to be flexible. $\triangledown$

Example 7.1.6 We now consider pure typed lambda-calculus.

- The members of the families R and F are empty, except:
  - $\mathsf{R}_{((\theta\to\theta')\times\theta)\theta'} = \{\mathbf{app}_{\theta\theta'}\}$;
  - $\mathsf{R}_{\theta^2\boldsymbol{\Omega}} = \{\approx_\theta\}$.
- All members of the family Q are empty, except:
  - $\mathsf{Q}_{\theta\theta'(\theta\to\theta')} = \{\lambda_{\theta\theta'}\}$.

Note that here we chose not to include flexible elements in the signature. $\triangledown$

Example 7.1.7 Finally, consider higher-order intuitionistic logic.

- The members of the families R and F are empty, except:
  - $\mathsf{R}_{((\theta\to\theta')\times\theta)\theta'} = \{\mathbf{app}_{\theta\theta'}\}$;
  - $\mathsf{R}_{\theta^2\boldsymbol{\Omega}} = \{\approx_\theta\}$.
- All members of the family Q are empty, except:
  - $\mathsf{Q}_{\theta\boldsymbol{\Omega}(\theta\to\boldsymbol{\Omega})} = \{\mathbf{set}_\theta\}$.

Note that we chose not to include flexible elements in this signature (that will be denoted by Σ_{HOL} in the sequel).

In this example, and in subsequent examples, we may omit the typing of the variables and other symbols when no confusion arises, writing $\approx$ for $\approx_\theta$ and so on. We may also use traditional infix notation, writing $(\gamma_1 \wedge \gamma_2)$ for $(\wedge\langle\gamma_1, \gamma_2\rangle)$, $\{x : \gamma\}$ for $(\mathbf{set} x\, \gamma)$, $t(t')$ for $(\mathbf{app}\langle t, t'\rangle)$ and so on. Finally, we may write simply f instead of $(f\langle\rangle)$ whenever $f \in \mathsf{F}_{\mathbf{1}\theta}$.

Using this signature for higher-order intuitionistic logic and the notational conventions mentioned above, other logical operations (True, False, the propositional connectives and the traditional higher-order quantifiers) can be introduced through abbreviations as usual (see for instance [18]):

- Equivalence: $(\delta_1 \Leftrightarrow \delta_2)$ for $(\delta_1 \approx_{\boldsymbol{\Omega}} \delta_2)$.
- True: $\mathbf{t}$ for $(\langle\rangle \approx_{\mathbf{1}} \langle\rangle)$.
- Conjunction: $(\delta_1 \wedge \delta_2)$ for $(\langle\delta_1, \delta_2\rangle \approx_{(\boldsymbol{\Omega}\times\boldsymbol{\Omega})} \langle\mathbf{t}, \mathbf{t}\rangle)$.
- Implication: $(\delta_1 \Rightarrow \delta_2)$ for $((\delta_1 \wedge \delta_2) \Leftrightarrow \delta_1)$.

- Universal quantification: $(\forall_\theta x_k^\theta\, \delta)$ for $(\{x_k^\theta \,:\, \delta\} \approx_{P(\theta)} \{x_k^\theta \,:\, \mathbf{t}\})$.
- False: $\mathbf{f}$ for $(\forall_{\mathbf{\Omega}} x_1^{\mathbf{\Omega}}\, x_1^{\mathbf{\Omega}})$.
- Negation: $(\neg\, \delta)$ for $(\delta \Rightarrow \mathbf{f})$.
- Disjunction: $(\delta_1 \vee \delta_2)$ for

$$(\forall_{\mathbf{\Omega}} x_i^{\mathbf{\Omega}}\, (((\delta_1 \Rightarrow x_i^{\mathbf{\Omega}}) \wedge (\delta_2 \Rightarrow x_i^{\mathbf{\Omega}})) \Rightarrow x_i^{\mathbf{\Omega}})),$$

  where $x_i^{\mathbf{\Omega}}$ is a variable of type $\mathbf{\Omega}$ not occurring free in $\langle \delta_1, \delta_2 \rangle$.
- Existential quantification: $(\exists_\theta x_k^\theta\, \delta)$ for

$$(\forall_{\mathbf{\Omega}} x_i^{\mathbf{\Omega}}\, (\forall_\theta x_k^\theta ((\delta \Rightarrow x_i^{\mathbf{\Omega}}) \Rightarrow x_i^{\mathbf{\Omega}}))),$$

  where $x_i^{\mathbf{\Omega}}$ is a variable of type $\mathbf{\Omega}$ not occurring free in δ.

∇

7.2 Higher-order Hilbert calculi

We now concentrate on making precise the notion of deduction system we want to work with. Such systems include in general inference rules with provisos. As already pointed out in Chapter 6, these provisos are in order because, when substituting schema variables, undesired interactions between binding operators and variables can occur. The following example recalls one of those undesirable interactions. Let φ be the formula

$$(\xi_1 \Rightarrow (\forall x\, \xi_1)).$$

This formula is a theorem of, say, intuitionistic first-order logic. Consider a substitution ρ such that x occurs free in $\rho(\xi_1)$. Then, $\rho(\varphi)$ is the formula

$$(\rho(\xi_1) \Rightarrow (\forall x\, \rho(\xi_1)))$$

which, in general, is not an intuitionistic theorem. Thus, a substitution applied to a theorem resulted in a formula that may no longer be a theorem. In order to avoid this undesired situation, we should state that $(\rho(\xi_1) \Rightarrow (\forall x\, \rho(\xi_1)))$ is obtained from $(\xi_1 \Rightarrow (\forall x\, \xi_1))$, *provided that* "$x$ is not free in $\rho(\xi_1)$". In other words, inference rules impose that just some substitutions are allowed when used in derivations. We thus start by defining what, in this context, we mean by a proviso as a "predicate" on substitutions.

Definition 7.2.1 By a *substitution* σ over Σ we mean a family

$$\sigma = \{\sigma_\theta\}_{\theta \in \Theta(\mathbf{S})}$$

such that $\sigma_\theta : \Xi_\theta \to T(\Sigma)_\theta$ is a map, for every $\theta \in \Theta(\mathbf{S})$. Analogously, a *ground substitution* ρ over Σ is a family $\rho = \{\rho_\theta\}_{\theta\in\Theta(\mathbf{S})}$ such that $\rho_\theta : \Xi_\theta \to gT(\Sigma)_\theta$ is a map, for every $\theta \in \Theta(\mathbf{S})$. $\triangledown$

Evidently every ground substitution is a substitution, but the converse is not necessarily true. As usual we write $\rho(t)$ and $\sigma(t)$ instead of $\hat{\rho}(t)$ and $\hat{\sigma}(t)$ for any $t \in T(\Sigma)$, where $\hat{\rho} : T(\Sigma) \to gT(\Sigma)$ and $\hat{\sigma} : T(\Sigma) \to T(\Sigma)$ are defined inductively from ρ and σ as expected (recall Definition 7.1.2). It is worthwhile to recall that $\hat{\rho}(\xi^x_{\xi'}) = (\rho(\xi))^x_{\rho(\xi')}$, where the right-side expression is the ground Σ-term obtained from $\rho(\xi)$ by replacing every free occurrence of x by $\rho(\xi')$. Analogously, $\hat{\sigma}(\xi^x_{\xi'}) = (\sigma(\xi))^x_{\sigma(\xi')}$. We denote by $Sbs(\Sigma)$ and $gSbs(\Sigma)$ the set of all substitutions and ground substitutions over Σ, respectively.

Definition 7.2.2 Let Σ be a signature in the sense of Definition 7.1.1. By a *local proviso* over Σ we mean a map $\pi : gSbs(\Sigma) \to 2$. $\triangledown$

Intuitively, $\pi(\rho) = 1$ if and only if the ground substitution ρ over Σ is allowed, and so π defines the set of ground substitutions allowed by the proviso. Recall the example above: $(\rho(\xi_1) \Rightarrow (\forall x\, \rho(\xi_1)))$ can be inferred from $(\xi_1 \Rightarrow (\forall x\, \xi_1))$ *provided that* "x is not free in $\rho(\xi_1)$". In this case the intended proviso is defined by: $\pi(\rho) = 1$ if and only if x is not free in $\rho(\xi_1)$. Thus, this proviso defines the set of ground substitutions ρ such that x is not free in $\rho(\xi_1)$ (note that this proviso depends on x and ξ_1).

This local notion of proviso is not sufficient for the purposes of fibring because we must be capable to translate rules from one signature to another. So, a universal notion of proviso that may be evaluated at any signature is needed. At this point, it is convenient to introduce the category **HSig** of signatures.

Definition 7.2.3 The category **HSig** of signatures is defined as follows: Its objects are signatures and given signatures $\Sigma = \langle \mathsf{R}, \mathsf{F}, \mathsf{Q}\rangle$ and $\Sigma' = \langle \mathsf{R}', \mathsf{F}', \mathsf{Q}'\rangle$, a signature morphism $h : \Sigma \to \Sigma'$ is a triple

$$h = \langle \{h^1_{\theta\theta'}\}_{\theta,\theta'\in\Theta(\mathbf{S})}, \{h^2_{\theta\theta'}\}_{\theta,\theta'\in\Theta(\mathbf{S})}, \{h^3_{\theta\theta'\theta''}\}_{\theta,\theta',\theta''\in\Theta(\mathbf{S})}\rangle$$

such that $h^1_{\theta\theta'} : \mathsf{R}_{\theta\theta'} \to \mathsf{R}'_{\theta\theta'}$, $h^2_{\theta\theta'} : \mathsf{F}_{\theta\theta'} \to \mathsf{F}'_{\theta\theta'}$ and $h^3_{\theta\theta'\theta''} : \mathsf{Q}_{\theta\theta'\theta''} \to \mathsf{Q}'_{\theta\theta'\theta''}$ are maps. Composition of signature morphisms and identity signature morphisms in **HSig** are defined pointwise, in the usual set-theoretic way. $\triangledown$

For simplicity, we will write $h(s)$ instead of $h^i_{\theta\theta'}(s)$ or $h^3_{\theta\theta'\theta''}(s)$ $(i = 1, 2)$ for a given symbol $s \in (\bigcup \mathsf{R}) \cup (\bigcup \mathsf{F}) \cup (\bigcup \mathsf{Q})$ of Σ.

For any signature morphism $h : \Sigma \to \Sigma'$ in **HSig** there is a unique extension $\hat{h} : T(\Sigma) \to T(\Sigma')$ of h preserving the operations, such that $\hat{h}(t) = t$ for $t \in \Xi_\theta \cup X_\theta$. Analogously, there is a unique extension $\hat{h} : gT(\Sigma) \to gT(\Sigma')$ of h (namely, the restriction of $\hat{h}$ to $gT(\Sigma)$). It is clear that, in **HSig**, the signature

$$\Sigma_1 = \langle \{\{r_{\theta\theta'}\}\}_{\theta,\theta'\in\Theta(\mathbf{S})}, \{\{f_{\theta\theta'}\}\}_{\theta,\theta'\in\Theta(\mathbf{S})}, \{\{q_{\theta\theta'\theta''}\}\}_{\theta,\theta',\theta''\in\Theta(\mathbf{S})}\rangle$$

with three families of singletons is terminal. We denote by $!_\Sigma$ the unique signature morphism from Σ to Σ_1.

Observe that any given local proviso π over Σ_1 can be easily extended to another signature Σ as follows: $\pi_\Sigma : gSbs(\Sigma) \to 2$ is given by

$$\pi_\Sigma(\rho) = \pi(\hat{!}_\Sigma \circ \rho).$$

Here, $\hat{!}_\Sigma \circ \rho$ denotes the ground substitution $\{\hat{!}_\Sigma \circ \rho_\theta\}_{\theta \in \Theta(\mathbf{S})}$ over Σ_1 obtained from the ground substitution ρ over Σ (see Figure 7.1).

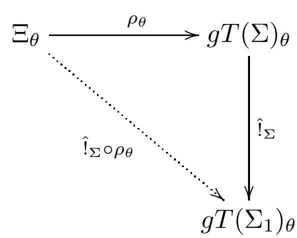


Figure 7.1: Extending a local proviso over Σ_1 to a signature Σ

From this, the following definition naturally arises:

Definition 7.2.4 A *universal proviso* (or, simply, a *proviso*) is a map

$$\Pi : gSbs(\Sigma_1) \to 2.$$

We denote by *Prov* the set of all universal provisos which includes the unit proviso $\mathbf{U}$ such that $\mathbf{U}(\rho) = 1$ for every ground substitution over Σ_1. $\triangledown$

Observe that provisos about binding are universal in the above sense, that is, they can be defined in the terminal signature and extended to other signatures. This is the case, for instance, for the proviso "x is not free in $\rho(\delta)$" (where δ is a formula), which generalizes the proviso given above as a motivating example.

We now introduce inference rules and the notion of Hilbert calculus.

Definition 7.2.5 A *inference rule* over Σ is a triple $\langle \Gamma, \delta, \Pi \rangle$ where $\Gamma \cup \{\delta\} \subseteq L(\Sigma)$ and Π is a (universal) proviso. $\triangledown$

For simplicity we may say "rule" instead of "inference rule". As before, if $\Gamma = \emptyset$ the conclusion δ of the rule is also known as an *axiom*. When the set Γ of premises is finite the rule is said to be *finitary*.

Definition 7.2.6 A *higher-order Hilbert calculus* is a triple

$$H = \langle \Sigma, R_g, R_\ell \rangle$$

where Σ is a signature and both R_g and R_ℓ are sets of finitary rules over Σ and $R_\ell \subseteq R_g$. ∇

To keep things simple, from now on "Hilbert calculus" will stand for "higher-order Hilbert calculus".

The elements of R_g are called *global rules* and those of R_ℓ are known as *local rules*. As we shall see, the former are the syntactical counterparts of global entailments, and the latter of local entailments. Naturally, deductions also appear in two forms: global and local derivations. This is exactly the same situation described in the previous chapters, but adapted to the present setting.

Before defining global and local derivations we need some further notation about provisos. If $\Pi, \Pi' \in Prov$ then the proviso $\Pi \sqcap \Pi' \in Prov$ is defined as follows: $(\Pi \sqcap \Pi')(\rho) = 1$ if and only if $\Pi(\rho) = \Pi'(\rho) = 1$. Furthermore, we say that $\Pi \leq \Pi'$ if and only if $\Pi = \Pi \sqcap \Pi'$. Finally, given $\Pi \in Prov$ and a substitution σ over Σ, we denote by $(\Pi\sigma)$ the (universal) proviso defined as follows:

$$(\Pi\sigma)(\rho) = \Pi(\hat{\rho} \circ \hat{!}_\Sigma \circ \sigma).$$

Here, $\hat{\rho} \circ \hat{!}_\Sigma \circ \sigma$ denotes the ground substitution $\{\hat{\rho} \circ \hat{!}_\Sigma \circ \sigma_\theta\}_{\theta \in \Theta(\mathbf{S})}$ over Σ_1 obtained from the ground substitution ρ over Σ (see Figure 7.2).

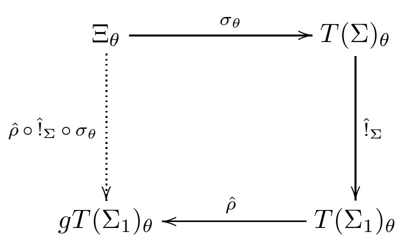


Figure 7.2: Defining proviso $(\Pi\sigma)$

The notion of context will be in order:

Definition 7.2.7 By a *context* we mean a finite sequence $\vec{x} = x_1 \dots x_n$ of distinct variables. ∇

We denote by [] the *empty context*. Given a context $\vec{x} = x_1 \dots x_n$ where the variables $x_1, \dots, x_n$ are of type $\theta_1, \dots, \theta_n$, respectively, we write $\theta_{\vec{x}}$ for $\theta_1 \times \dots \times \theta_n$ and say that $\theta_{\vec{x}}$ is the type of the context $\vec{x}$. This convention is obviously extended to the empty context: $\theta_{[]}$ is $\mathbf{1}$.

Given a set of terms using a finite number of free variables, we may refer to its *canonical context* formed exclusively by those free variables (this canonical context is unique once we fix a total ordering of the variables).

From now on we will use the families of sets $T(\Sigma, \vec{x})$, $gT(\Sigma, \vec{x})$ and the sets $L(\Sigma, \vec{x})$, $gL(\Sigma, \vec{x})$ with the obvious meanings: In each case we use only variables in the indicated context $\vec{x}$.

Definition 7.2.8 A *global $\vec{x}$-derivation* within a Hilbert calculus H of $\delta \in L(\Sigma, \vec{x})$ from $\Gamma \subseteq L(\Sigma, \vec{x})$ with proviso Π is a sequence of pairs $\langle \delta_1, \Pi_1 \rangle \dots \langle \delta_n, \Pi_n \rangle$ in $L(\Sigma, \vec{x}) \times \mathit{Prov}$ such that δ_n is δ, Π_n is Π and for each $i = 1, \dots, n$:

- either $\delta_i \in \Gamma$ and Π_i is arbitrary;
- or there are a rule $\langle \{\gamma'_1, \dots, \gamma'_k\}, \delta', \Pi' \rangle \in R_g$ and a substitution σ over Σ such that:
  - for each $j = 1, \dots, k$, there is a $i_j \in \{1, \dots, i-1\}$ such that $\delta_{i_j} = \sigma(\gamma'_j)$;
  - $\delta_i = \sigma(\delta')$;
  - $\Pi_i \leq \Pi_{i_1} \sqcap \dots \sqcap \Pi_{i_k} \sqcap (\Pi'\sigma)$.

We write $\Gamma \vdash_H^{g\vec{x}} \delta \lhd \Pi$ when there is such a global $\vec{x}$-derivation in H of δ from Γ with proviso Π. And we use the notation $\Gamma \vdash_H^{g} \delta \lhd \Pi$ to indicate that $\Gamma \vdash_H^{g\vec{x}} \delta \lhd \Pi$ for some context $\vec{x}$. ▿

Definition 7.2.9 A *local $\vec{x}$-derivation* within a Hilbert calculus H of $\delta \in L(\Sigma, \vec{x})$ from $\Gamma \subseteq L(\Sigma, \vec{x})$ with proviso Π is a sequence of pairs $\langle \delta_1, \Pi_1 \rangle \dots \langle \delta_n, \Pi_n \rangle$ in $L(\Sigma, \vec{x}) \times \mathit{Prov}$ such that δ_n is δ, Π_n is Π and for each $i = 1, \dots, n$:

- either $\delta_i \in \Gamma$ and Π_i is arbitrary;
- or $\emptyset \vdash_H^{g\vec{x}} \delta_i \lhd \Pi_i$;
- or there are a rule $\langle \{\gamma'_1, \dots, \gamma'_k\}, \delta', \Pi' \rangle \in R_\ell$ and a substitution σ over Σ such that:
  - for each $j = 1, \dots, k$, there is a $i_j \in \{1, \dots, i-1\}$ such that $\delta_{i_j} = \sigma(\gamma'_j)$;
  - $\delta_i = \sigma(\delta')$;
  - $\Pi_i \leq \Pi_{i_1} \sqcap \dots \sqcap \Pi_{i_k} \sqcap (\Pi'\sigma)$.

When there is such a local $\vec{x}$-derivation in H of δ from Γ with proviso Π, we write $\Gamma \vdash_H^{\ell\vec{x}} \delta \lhd \Pi$. And when there is a context $\vec{x}$ such that $\Gamma \vdash_H^{\ell\vec{x}} \delta \lhd \Pi$ we write $\Gamma \vdash_H^{\ell} \delta \lhd \Pi$. ▿

As usual with respect to both global and local derivations, we may omit the reference to the assumptions when $\Gamma = \emptyset$. The reference to the Hilbert calculus may also be omitted when obvious, and furthermore, when $\Pi = \mathbf{U}$, we may also omit the reference to the proviso.

We conclude this section with two interesting examples.

Example 7.2.10 We consider again modal propositional logic. Let Σ be a signature as described in Example 7.1.3. We establish the deductive component of the modal logic **K** by endowing it with the set R_ℓ composed of the following rules:

(taut1) $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)), \mathbf{U}\rangle$;

(taut2) $\langle\emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))), \mathbf{U}\rangle$;

(taut3) $\langle\emptyset, (((\neg\,\xi_1) \Rightarrow (\neg\,\xi_2)) \Rightarrow (((\neg\,\xi_1) \Rightarrow \xi_2) \Rightarrow \xi_1))), \mathbf{U}\rangle$;

(norm) $\langle\emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2))), \mathbf{U}\rangle$;

(MP) $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2, \mathbf{U}\rangle$;

and the set R_g containing the rules in R_ℓ plus necessitation:

(Nec) $\langle\{\xi_1\}, (\Box\xi_1), \mathbf{U}\rangle$. ▽

Example 7.2.11 We consider again higher-order intuitionistic logic. Let Σ_{HOL} be the signature defined in Example 7.1.7. We need to introduce some additional notation for provisos:

- $x \prec \delta$ denotes the (universal) proviso that, for each $\rho \in gSbs(\Sigma_1)$, returns the value of the assertion "x occurs free in $\rho(\delta)$" (note that x and δ are fixed);

- $x \not\prec \delta$ denotes the (universal) proviso that, for each $\rho \in gSbs(\Sigma_1)$, returns the value of the assertion "x does not occur free in $\rho(\delta)$" (note that x and δ are fixed);

- $\delta_1 \rhd x : \delta_2$ denotes the (universal) proviso that, for each $\rho \in gSbs(\Sigma_1)$, returns the value of the assertion "$\rho(\delta_1)$ is free for x in $\rho(\delta_2)$" (note that x, δ_1 and δ_2 are fixed).

From now on we will use the set-theoretic abbreviations in the context of higher-order logic used in [18]. Thus, U_θ stands for the set $\{x^\theta \ : \ \mathbf{t}\}$ and $t_2^{t_1}$ stands for the set

$$\{h \subseteq t_1 \times t_2 \ : \ (\forall x((x \in t_1) \Rightarrow (\exists! y((y \in t_2) \wedge (\langle x, y\rangle \in h)))))\}.$$

The deductive component of the envisaged higher-order logic is as follows (omitting the types of schema variables, variables and other symbols and assuming that $i \in \mathbb{N}$, $k \geq 2$ and θ, $\theta_1, \ldots, \theta_k$ are types):

- R_ℓ is the set composed by:

  (taut1) $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)), \mathbf{U}\rangle$;

  (taut2) $\langle\emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))), \mathbf{U}\rangle$;

  (taut3) $\langle\emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_1 \Rightarrow \xi_3) \Rightarrow (\xi_1 \Rightarrow (\xi_2 \wedge \xi_3)))), \mathbf{U}\rangle$;

  (taut4) $\langle\emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow (\xi_1 \wedge \xi_2))), \mathbf{U}\rangle$;

(uni) $\langle\emptyset, (\forall x_1(x_1 \approx \langle\rangle)), \mathbf{U}\rangle$;
(equa$_{i,\theta}$) $\langle\emptyset, ((\xi_1 \approx \xi_2) \Rightarrow (\xi_3{}^{x_i}_{\xi_1} \Rightarrow \xi_3{}^{x_i}_{\xi_2})), (\xi_1 \rhd x_i : \xi_3) \sqcap (\xi_2 \rhd x_i : \xi_3)\rangle$;
(ref$_\theta$) $\langle\emptyset, (\forall x_1(x_1 \approx x_1)), \mathbf{U}\rangle$;
(proj$_{k,\theta_1,\ldots,\theta_k,i}$) $\langle\emptyset, (\forall x_1 \cdots \forall x_k((\langle x_1, \ldots, x_k\rangle)_i \approx x_i)), \mathbf{U}\rangle$ for $1 \leq i \leq k$;
(prod$_{k,\theta_1,\ldots,\theta_k}$) $\langle\emptyset, (\forall x_1(x_1 \approx \langle(x_1)_1, \ldots, (x_1)_k\rangle)), \mathbf{U}\rangle$;
(comph$_\theta$) $\langle\emptyset, (\forall x_1(x_1 \in \{x_1 : \xi_1\} \Leftrightarrow \xi_1)), \mathbf{U}\rangle$;
(subs$_{i,\theta}$) $\langle\emptyset, ((\forall x_i\xi_2) \Rightarrow \xi_2{}^{x_i}_{\xi_1}), ((\xi_1 \rhd x_i : \xi_2) \sqcap (x_i \prec \xi_2))\rangle$;
(fun$_{\theta,\theta'}$) $\langle\emptyset, \forall x_1(x_1 \in U_{\theta'}^{U_\theta} \Rightarrow \exists! x_2 \forall x_3 \forall x_4(\langle x_3, x_4\rangle \in x_1 \Leftrightarrow x_2(x_3) \approx x_4)), \mathbf{U}\rangle$;
(equiv) $\langle\emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_2 \Rightarrow \xi_1) \Rightarrow (\xi_1 \Leftrightarrow \xi_2))), \mathbf{U}\rangle$;
(MP) $\langle\{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2, \mathbf{U}\rangle$;

- R_g is obtained by adding to R_ℓ the following rules:

  (Gen$_{i,\theta}$) $\langle\{(\xi_1 \Rightarrow \xi_2)\}, (\xi_1 \Rightarrow (\forall x_i\xi_2)), x_i \not\prec \xi_1\rangle$. ▽

Remark 7.2.12 The Hilbert calculus above is an adaptation of the sequent calculus S for higher-order logic presented in [18] under the name of local set theory. In fact, the present Hilbert calculus generalizes local set theory because of the use of types of the form $(\theta \to \theta')$. More specifically, axiom $\mathbf{fun}_{\theta,\theta'}$ generalizes the extensionality axiom of local set theory. In [63] it is shown that

$$(x_1 \approx x_1), \ldots, (x_n \approx x_n), \Psi \vdash_S \varphi$$

is provable in S if and only if $\Psi \vdash^{\ell\vec{x}} \varphi$ in the Hilbert calculus above, provided that every formula in $\Psi \cup \{\varphi\}$ is written in the language of [18]; that is, provided that $\theta' = \mathbf{\Omega}$ in any type of the form $(\theta \to \theta')$. ▽

We will prove below (see Proposition 7.5.4) a general completeness theorem for the Hilbert calculus of Example 7.2.11 with respect to an appropriate topos-theoretic semantics.

7.3 Higher-order interpretation systems

When working with higher-order based logics, it is natural to adopt a topos theoretic semantics in the style of categorical logic (see for instance [221]).

However, a slight generalization is needed in order to fulfill the second goal discussed at the beginning of Section 7.1. Indeed, given a Kripke semantics for an arbitrary modality we would like to be able to generate the corresponding topos semantics while preserving the original models in a precise sense: Each of the original models should be converted into a topos model with the same denotation of terms and formulas.

Since we wanted to be able to cope with arbitrary modal-like operators, we were led to a more general topos semantics achieved by endowing each model with an extra parameter (an object W of the topos) playing the role of the world space. As shown in some examples at this section, this generalization is effective with respect to the issue at hand: denotation of terms/formulas is preserved when obtaining a topos model from a model in a given logic. Hence, entailment is also preserved. Furthermore, the extra parameter also allows an explicit distinction between modal and other operators, which helps the intuitions of a reader more familiarized with traditional semantics.

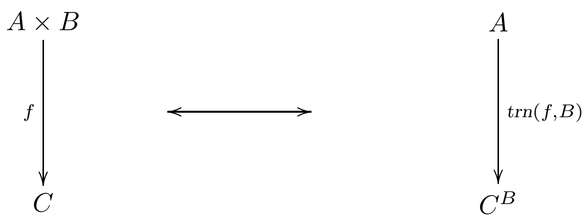


Figure 7.3: Exponential transpose of f

It should be stressed that the extra parameter is not necessary for achieving completeness (as proved in Section 7.5). It is only necessary for being able to generate a topos model from each given Kripke model preserving the denotation of terms and formulas. As explained before, this is essential in order to make the homogeneous scenario of fibring more useful for applications.

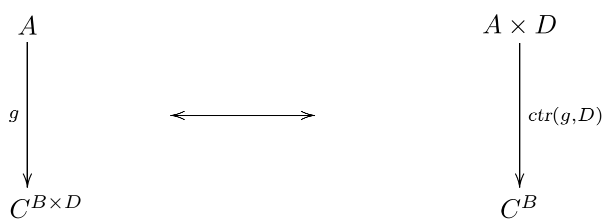


Figure 7.4: Exponential cotranspose of g

From now on, we use the following notation in the context of topos theory. If $f : A \times B \to C$ is a morphism in a given topos then $trn(f, B) : A \to C^B$ is the exponential transpose of f obtained from the definition of C^B (see Figure 7.3).

If $g : A \to C^{B \times D}$ is a morphism in a topos then $ctr(g, D) : A \times D \to C^B$ is the exponential cotranspose of g obtained from the definition of $(C^B)^D$ and the canonical isomorphism between $(C^B)^D$ and $C^{B\times D}$ (see Figure 7.4).

Finally, we refer the extent (or support) of an object A in a topos. The extent of A, denoted by $E(A)$, is the (domain of the) subobject of 1 given by the direct image $\exists_{!_A}(\mathrm{id}_A)$ of A along $!_A$, where $!_A$ is the unique morphism from the object A to the terminal object 1 (see Figure 7.5).

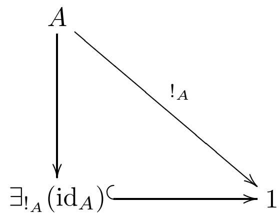


Figure 7.5: Extent of an object A

Definition 7.3.1 An *interpretation structure* over Σ is a triple $M = \langle \mathcal{E}, W, \cdot_M \rangle$ where $\mathcal{E}$ is a topos, W is an object of $\mathcal{E}$ such that $E(W) = 1$, and $\cdot_M$ is an interpretation map satisfying the following properties:

- for $\theta \in \Theta(\mathbf{S})$, θ_M is an object of $\mathcal{E}$ such that:
  - $\mathbf{1}_M$ is terminal;
  - $\mathbf{\Omega}_M$ is a subobject classifier Ω;
  - $(\theta_1 \times \cdots \times \theta_n)_M = \theta_{1M} \times \cdots \times \theta_{nM}$;
  - $(\theta \to \theta')_M = (\theta'_M)^{\theta_M}$;
- for $r \in \mathsf{R}_{\theta\theta'}$,
  - $r_M = \{r_{M\tau}\}_{\tau \in \Theta(\mathbf{S})}$ where $r_{M\tau} \in \mathcal{E}((\theta_M)^{\tau_M}, ({\theta'}_M)^{\tau_M})$ (the set of morphisms in $\mathcal{E}$ from $(\theta_M)^{\tau_M}$ to $({\theta'}_M)^{\tau_M}$)). The family r_M must be natural in the following sense: Given $\tau, \tau' \in \Theta(\mathbf{S})$ and $m \in \mathcal{E}(W \times \tau_M, W \times {\tau'}_M)$, $n \in \mathcal{E}(W \times {\tau'}_M, \theta_M)$, then

$$ctr(r_{M\tau'} \circ trn(n, {\tau'}_M), {\tau'}_M) \circ m = ctr(r_{M\tau} \circ trn(n \circ m, \tau_M), \tau_M);$$

- for $f \in \mathsf{F}_{\theta\theta'}$,
  - $f_M = \{f_{M\tau}\}_{\tau \in \Theta(\mathbf{S})}$ where $f_{M\tau} \in \mathcal{E}((\theta_M)^{W \times \tau_M}, ({\theta'}_M)^{W \times \tau_M})$. The family f_M must be natural in the following sense: Given $\tau, \tau' \in \Theta(\mathbf{S})$ and $m \in \mathcal{E}(W \times \tau_M, W \times {\tau'}_M)$, $n \in \mathcal{E}(W \times {\tau'}_M, \theta_M)$, then

$$ctr(f_{M\tau'} \circ trn(n, W \times \tau'_M), W \times \tau'_M) \circ m$$
$$=$$
$$ctr(f_{M\tau} \circ trn(n \circ m, W \times \tau_M), W \times \tau_M);$$

- for $q \in \mathsf{Q}_{\theta\theta'\theta''}$,
  - $q_M = \{q_{M\tau}\}_{\tau \in \Theta(\mathbf{S})}$ where $q_{M\tau} \in \mathcal{E}((\theta'_M)^{\tau_M \times \theta_M}, (\theta''_M)^{\tau_M})$. The family q_M must be natural in the following sense: Given $\tau, \tau' \in \Theta(\mathbf{S})$ and $m \in \mathcal{E}(W \times \tau_M, W \times \tau'_M)$, $n \in \mathcal{E}(W \times \tau'_M \times \theta_M, \theta'_M)$, then

$$ctr(q_{M\tau'} \circ trn(n, \tau'_M \times \theta_M), \tau'_M) \circ m$$
$$=$$
$$ctr(q_{M\tau} \circ trn(n \circ (m \times \mathrm{id}_{\theta_M}), \tau_M \times \theta_M), \tau_M).$$

We denote by $Str(\Sigma)$ the class of all interpretation structures over Σ. ∇

The naturality requirements for the rigid, flexible and binding operators are related to the *Substitution Lemma*, as we shall see below in Remark 7.3.3 and Proposition 7.3.4.

Once we have the notion of interpretation structure over Σ, the following step is to define the denotation of terms in a given interpretation structure over Σ. Since we add modalities to the languages, it will be necessary to slightly generalize the usual notion of categorical semantics. This generalization will be shown to be appropriate in Proposition 7.5.4 below.

Definition 7.3.2 Let $\vec{x} = x_1 \dots x_n$ be a context with type $\theta_{\vec{x}} = \theta_1 \times \cdots \times \theta_n$, and $\theta_{\vec{x}M} = \theta_{1M} \times \cdots \times \theta_{nM}$. Then, the *denotation map*

$$[\![\cdot]\!]^M_{\vec{x}} : gT(\Sigma, \vec{x})_{\theta'} \to \mathcal{E}(W \times \theta_{\vec{x}M}, \theta'_M)$$

of ground terms of type θ' with free variables in $\vec{x}$ is inductively defined as follows:

- $[\![x_i]\!]^M_{\vec{x}} = p_i$ where p_i is the projection from $W \times \theta_{\vec{x}M}$ to θ_{iM};
- $[\![r\,t]\!]^M_{\vec{x}} = ctr(r_{M\theta_{\vec{x}}} \circ trn([\![t]\!]^M_{\vec{x}}, \theta_{\vec{x}M}), \theta_{\vec{x}M})$;
- $[\![f\,t]\!]^M_{\vec{x}} = ctr(f_{M\theta_{\vec{x}}} \circ trn([\![t]\!]^M_{\vec{x}}, W \times \theta_{\vec{x}M}), W \times \theta_{\vec{x}M})$;

- $[\![qx\,t]\!]^M_{\vec{x}} = ctr(q_{M\theta_{\vec{x}}} \circ trn([\![t^x_y]\!]^M_{\vec{x}y}, \theta_{\vec{x}M} \times \theta_M), \theta_{\vec{x}M})$ where y is of the same type θ as x and does not occur in $\vec{x}$;
- $[\![\langle\rangle]\!]^M_{\vec{x}} = !_{W\times\theta_{\vec{x}M}}$;
- $[\![\langle t_1, \ldots, t_k\rangle]\!]^M_{\vec{x}} = ([\![t_1]\!]^M_{\vec{x}}, \ldots, [\![t_k]\!]^M_{\vec{x}})$ for $k \geq 2$;
- $[\![(t)_i]\!]^M_{\vec{x}} = p_i \circ [\![t]\!]^M_{\vec{x}}$ where p_i is the projection from $\theta_{1M} \times \cdots \times \theta_{kM}$ to θ_{iM}. ∇

Remark 7.3.3 In order to better understand the naturality requirement for the family r_M (for $r \in \mathsf{R}_{\theta\theta'}$, a rigid function symbol according to Definition 7.3.1), we must think about r_M as representing a morphism $m_r : \theta_M \to \theta'_M$ in every possible context (with type) τ.

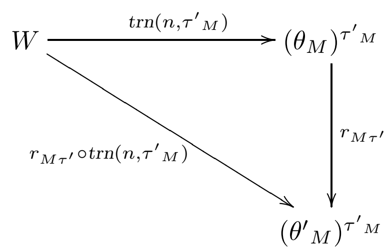


Figure 7.6: Interpretation of terms

The idea is that m_r realizes the ground term $r(x)$ in the interpretation structure M, and the component $r_{M\tau}$ of r_M constitutes the morphism m_r in context τ. Now, given a tuple of ground terms $\langle t_1, \ldots, t_k\rangle$ (interpreted in a given context τ as a morphism m)[1] as well as a ground term $t(x_1, \ldots, x_k)$ (interpreted in a context τ' as a morphism n) such that m and n are composable (that is, such that there exists the morphism $n \circ m$ interpreting the ground term $t^{x_1\cdots x_k}_{t_1\cdots t_k}$ in context τ) then the interpretation of the ground terms $r(t)^{x_1\cdots x_k}_{t_1\cdots t_k}$ and $r(t^{x_1\cdots x_k}_{t_1\cdots t_k})$ in context τ must coincide.

That is, the substitution lemma must hold good in the proposed semantic framework. Let us see this using diagrams: Given $n : W \times \tau'_M \to \theta_M$ consider the diagram in Figure 7.6 and define

$$l = ctr(r_{M\tau'} \circ trn(n, \tau'_M), \tau'_M) : W \times \tau'_M \to \theta'_M.$$

The morphism l is the interpretation in M of $r(t)$ (that is, r^x_t) in context τ'.

On the other hand, starting from $n \circ m$ in Figure 7.7 (the interpretation in M of $t^{x_1\cdots x_k}_{t_1\cdots t_k}$ in context τ) consider the composition described in the diagram in Figure 7.8 and let

$$q = ctr(r_{M\tau} \circ trn(n \circ m, \tau_M), \tau_M) : W \times \tau_M \to \theta'_M$$

[1] A slight simplification was made here in order to explain the basic idea; see the right formulation in Proposition 7.3.4 below.

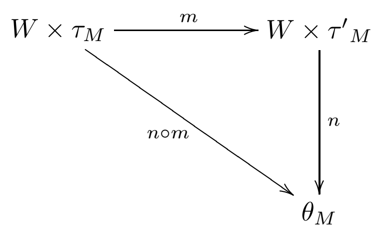


Figure 7.7: Morphism $n \circ m$

be the interpretation in M of $r(t^{x_1 \cdots x_k}_{t_1 \cdots t_k})$ in context τ.

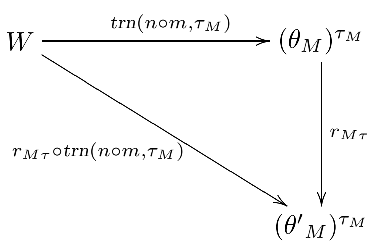


Figure 7.8: Morphism $r_{M\tau} \circ trn(n \circ m, \tau_M)$

Then, the naturality condition requires that the diagram in Figure 7.9 commutes, where $l \circ m$ is the interpretation in M of $r(t)^{x_1 \cdots x_k}_{t_1 \cdots t_k}$ in context τ.

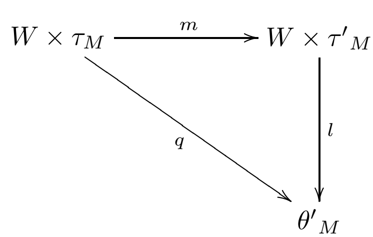


Figure 7.9: Naturality condition

The naturality requirements for the interpretation of flexible and (binding) operator symbols can be analyzed analogously, taking into account the appropriate modifications required in each case. ∇

As already referred above, the naturality of the families r_M (for $r \in \mathsf{R}_{\theta\theta'}$), f_M (for $f \in \mathsf{F}_{\theta\theta'}$) and q_M (for $q \in \mathsf{Q}_{\theta\theta'\theta''}$) allows us to prove (by a straightforward induction) the substitution lemma for interpretation structures over Σ.

Proposition 7.3.4 *Let t' be a ground term free for a variable x in a ground term t. Then, for any interpretation structure M over Σ and appropriate contexts $\vec{y}$ and $\vec{z}$, it holds:*

$$[\![t^x_{t'}]\!]^M_{\vec{y}\vec{z}} = [\![t]\!]^M_{\vec{y}x} \circ (\pi, [\![t']\!]^M_{\vec{y}\vec{z}})$$

where $\pi : W \times \Theta_{\vec{y}M} \times \Theta_{\vec{z}M} \to W \times \Theta_{\vec{y}M}$ is the canonical projection.

Using the concepts already defined, we can now introduce the notion of entailment (actually, of two entailments) for a given class of interpretation structures over Σ. Previous to this, we shall need more notation from topos theory. Given a morphism $\chi : A \to \Omega$ we denote by $mon(\chi) : dom(mon(\chi)) \to A$ (see Figure 7.10) the monomorphism obtained in the pullback of the diagram determined by $\{\chi, true\}$.

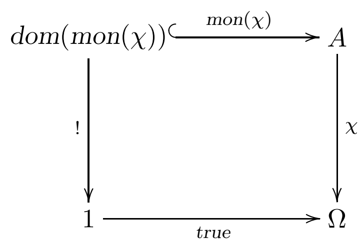


Figure 7.10: Monomorphism $mon(\chi)$

Given an object A of a topos $\mathcal{E}$, we denote by $Sub(A)$ the lattice of (equivalence classes of) subobjects of A.[2] The order in $Sub(A)$ is given as follows: $[f] \leq [g]$ if and only if there exists a (necessarily unique) arrow $h : dom(f) \to dom(g)$ in $\mathcal{E}$ such that $f = g \circ h$, that is, such that the diagram in Figure 7.11 commutes.

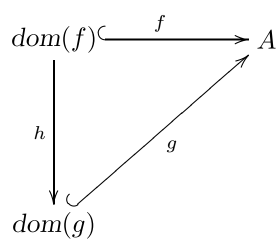


Figure 7.11: Order in $Sub(A)$

The order in $Sub(A)$ is well-defined. If $\chi_1, \chi_2 : A \to \Omega$ then we define $\chi_1 \leq \chi_2$ if and only if $[mon(\chi_1)] \leq [mon(\chi_2)]$. This relation is also well-defined. For each

[2]Recall (see [187]) that two monomorphisms $f : dom(f) \hookrightarrow A$ and $g : dom(g) \hookrightarrow A$ are *equivalent* if and only if there exists a (necessarily unique) isomorphism $h : dom(f) \to dom(g)$ in $\mathcal{E}$ such that $f = g \circ h$ (and then $g = f \circ h^{-1}$).

object A, $true_A$ denotes the arrow $true \circ !_A : A \to \Omega$ and $\bigwedge$ denotes the infimum in the lattice $Sub(A)$. Observe that the subobject of A associated to $true_A$ is $[\mathrm{id}_A]$, the top element of $Sub(A)$. Finally, given a monomorphism $f : dom(f) \to A$, we will denote by $char(f) : A \to \Omega$ the unique morphism such that $[mon(char(f))] = [f]$. Note that $char(mon(\chi)) = \chi$ for every $\chi : A \to \Omega$.

By Definition 7.3.2, if φ is a ground formula, $\vec{x}$ is a context for φ and M is an interpretation structure over Σ then

$$[\![\varphi]\!]^M_{\vec{x}} : W \times \theta_{\vec{x}M} \to \Omega$$

represents a subobject of $W \times \theta_{\vec{x}M}$, that is, an element of $Sub(W \times \theta_{\vec{x}M})$. This suggests the following definitions.

Definition 7.3.5 A *higher-order interpretation system* is a pair $\mathcal{I} = \langle \Sigma, \mathcal{M} \rangle$ where Σ is a signature and $\mathcal{M}$ is a class of interpretation structures over Σ. $\triangledown$

As was done before, from now on "interpretation system" will stand for "higher-order interpretation system".

Definition 7.3.6 Let $\mathcal{I}$ be an interpretation system and let $\Psi \cup \{\varphi\} \subseteq gL(\Sigma, \vec{x})$ be a finite set.

- Ψ *globally $\vec{x}$-entails* φ within $\mathcal{I}$, written $\Psi \vDash^{g\vec{x}}_{\mathcal{I}} \varphi$, if, for every $M \in \mathcal{M}$, $\bigwedge\limits_{\psi \in \Psi} [\![\psi]\!]^M_{\vec{x}} = true_{W \times \theta_{\vec{x}M}}$ implies $[\![\varphi]\!]^M_{\vec{x}} = true_{W \times \theta_{\vec{x}M}}$;
- Ψ *globally entails* φ within $\mathcal{I}$, written $\Psi \vDash^{g}_{\mathcal{I}} \varphi$, if $\Psi \vDash^{g\vec{x}}_{\mathcal{I}} \varphi$ choosing for $\vec{x}$ the canonical context of $\Psi \cup \{\varphi\}$;
- Ψ *locally $\vec{x}$-entails* φ within $\mathcal{I}$, written $\Psi \vDash^{\ell\vec{x}}_{\mathcal{I}} \varphi$, if, for every $M \in \mathcal{M}$, $\bigwedge\limits_{\psi \in \Psi} [\![\psi]\!]^M_{\vec{x}} \leq [\![\varphi]\!]^M_{\vec{x}}$;
- Ψ *locally entails* φ within $\mathcal{I}$, written $\Psi \vDash^{\ell}_{\mathcal{I}} \varphi$, if $\Psi \vDash^{\ell\vec{x}}_{\mathcal{I}} \varphi$ choosing for $\vec{x}$ the canonical context of $\Psi \cup \{\varphi\}$. $\triangledown$

Since the set Ψ in Definition 7.3.6 is finite, the notions of global and local entailment are well-defined: It is only required to compute the infimum

$$\bigwedge_{\psi \in \Psi} [\![\psi]\!]^M_{\vec{x}}$$

in the Heyting algebra $Sub(W \times \theta_{\vec{x}M})$, and compare it with $[\![\varphi]\!]^M_{\vec{x}}$. The finiteness of Γ guarantees that this infimum always exists. It is not possible to define this notion of entailment for arbitrary sets Ψ, because the lattices of the form $Sub(A)$ are not complete in general.

It should be noticed that the semantic notions introduced in Definition 7.3.6 correspond to the notions of global and local entailment defined in the previous chapters, adapted to the present setting. Moreover, the local entailment defined above coincides with the traditional notion of entailment in categorical logic. On the other hand, the global entailment proposed above brings to the topos setting the notion of global entailment usual in modal logic.

It is straightforward to prove that for any finite set $\Psi \cup \{\varphi\} \subseteq gL(\Sigma, \vec{x})$: $\Psi \vDash^{g}_{\mathcal{I}} \varphi$ implies $\Psi \vDash^{g\vec{x}}_{\mathcal{I}} \varphi$; and $\Psi \vDash^{\ell}_{\mathcal{I}} \varphi$ implies $\Psi \vDash^{\ell\vec{x}}_{\mathcal{I}} \varphi$. The converses are not necessarily true, because empty (that is, initial) domains are allowed in the interpretation of types.

To what concerns these entailments, we may drop the reference to the assumptions when $\Psi = \emptyset$, and may also omit the reference to the interpretation system.

As already mentioned, the following examples (modal propositional logic and first-order logic) show that for many well-known logics it is possible to lift the original semantics given to those logics to the topos semantics level, while preserving the two entailments. For such logics, working with the original semantics or with the proposed topos semantics are equivalent tasks. For this reason, not much generality is lost by assuming from now on that the logics we are handling are endowed with the topos semantics.

Previous to analyze the examples of modal and first-order logics, observe that we have substituted in Definition 7.3.1 the usual (set-theoretic) condition $W \neq \emptyset$ for $E(W) = 1$. The new requirement, which is stronger than the condition "W is not initial" (an apparently natural generalization to topos semantics of the set-theoretic requirement $W \neq \emptyset$),[3] is justified by the following:

Proposition 7.3.7 *Let A and B be objects in a given topos $\mathcal{E}$ such that $E(B) = 1$. Let $[f], [g] \in Sub(A)$. Then:*

$$[f] \leq [g] \quad \textit{if and only if} \quad [f \times id_B] \leq [g \times id_B].$$

Proof. It is a direct consequence of Proposition 2.9 in [58]. ◁

Example 7.3.8 We consider modal propositional logic. Let Σ be a signature as described in Example 7.1.3. Assume we are given a class $\mathcal{K}$ of general Kripke structures for Σ of the form $K = \langle W, R, \mathcal{B}, V \rangle$. This class defines the global and local entailments as usual. Very briefly, recall that $[\![\varphi]\!]^K \in \mathcal{B}$ is the admissible set of worlds where φ holds and:

- $\Gamma \vDash^{g}_{\mathcal{K}} \varphi$ if, for every $K \in \mathcal{K}$, $\displaystyle\bigcap_{\gamma \in \Gamma} [\![\gamma]\!]^K = W$ implies $[\![\varphi]\!]^K = W$;
- $\Gamma \vDash^{\ell}_{\mathcal{K}} \varphi$ if, for every $K \in \mathcal{K}$, $\displaystyle\bigcap_{\gamma \in \Gamma} [\![\gamma]\!]^K \subseteq [\![\varphi]\!]^K$.

[3]Notice that, in the topos of sets, it holds: W is not initial if and only if $W \neq \emptyset$ if and only if $E(W) = 1$. In an arbitrary (non-degenerate) topos, $E(W) = 1$ implies that W is not initial, but the converse is not true in general.

The idea is to generate from $\mathcal{K}$ a class of interpretation structures $\mathcal{M}_{\mathcal{K}}$ over Σ and check whether we recover the original entailments with the topos semantics.

For each $K \in \mathcal{K}$, let M_K be the interpretation structure $\langle \mathbf{Set}, W, \cdot_{M_K} \rangle$ over Σ such that:

- $\mathbf{f}_{M_K \tau} = \lambda h. (\lambda \vec{a}. 0)$;
- $\neg_{M_K \tau} = \lambda h. (\lambda \vec{a}. -h(\vec{a}))$;
- $\wedge_{M_K \tau} = \lambda h. (\lambda \vec{a}. (p_1(h(\vec{a})) \sqcap p_2(h(\vec{a}))))$;
- $\vee_{M_K \tau} = \lambda h. (\lambda \vec{a}. (p_1(h(\vec{a})) \sqcup p_2(h(\vec{a}))))$;
- $\Rightarrow_{M_K \tau} = \lambda h. (\lambda \vec{a}. (p_1(h(\vec{a})) \Rrightarrow p_2(h(\vec{a}))))$;
- $p_{M_K \tau} = \lambda h. (\lambda u \vec{a}. V_p(u))$ for any $p \in \mathbb{P}$;
- $\Diamond_{M_K \tau} = \lambda h. (\lambda u \vec{a}. \displaystyle\bigvee_{v \in W:\, uRv} h(v, \vec{a}))$;
- $\Box_{M_K \tau} = \lambda h. (\lambda u \vec{a}. \displaystyle\bigwedge_{v \in W:\, uRv} h(v, \vec{a}))$.

Here $-$, $\sqcap$, $\sqcup$ and $\Rrightarrow$ denote the operations of complement, meet, join and relative complement in the Boolean algebra *2*, respectively (recall Example 3.1.2); $p_1, p_2 : 2 \times 2 \to 2$ are the canonical projections, and $\bigvee, \bigwedge$ denote the joins and meets of subsets of 2. It is easy to verify that each of these families is natural in the sense of Definition 7.3.1. $\triangledown$

It is not hard to prove by induction on the complexity of the formula φ (identifying W with $W \times 1$) that $char([\![\varphi]\!]^K) = [\![\varphi]\!]^{M_K}$. Then, if we define

$$\mathcal{M}_{\mathcal{K}} = \{M_K : K \in \mathcal{K}\}$$

and

$$\mathcal{I}_{\mathcal{K}} = \langle \Sigma, \mathcal{M}_{\mathcal{K}} \rangle$$

it is straightforward to prove the following:

Proposition 7.3.9 *Let $\mathcal{K}$ be a class of Kripke structures for Σ. Then:*

- $\Gamma \vDash^g_{\mathcal{K}} \varphi$ *if and only if* $\Gamma \vDash^g_{\mathcal{I}_{\mathcal{K}}} \varphi$*;*
- $\Gamma \vDash^\ell_{\mathcal{K}} \varphi$ *if and only if* $\Gamma \vDash^\ell_{\mathcal{I}_{\mathcal{K}}} \varphi$*.*

Example 7.3.10 We now consider modal intuitionistic propositional logic. Because of the intuitionistic character of the internal logic of topoi, it seems natural to consider the intuitionistic version of modal logic in the present framework. Consider the signature Σ_{int} obtained from the signature Σ described in Example 7.1.3 by making the following changes: The symbols $\neg$ and $\Rightarrow$ are now flexible. Consider a class $\mathcal{K}_{int}$ of general Kripke structures for Σ_{int} of the form

$$K = \langle W, \leq, R, \mathcal{B}, V \rangle$$

as well as the usual notions of global and local entailments. Note that $u \leq v$ and $V_p(u) = 1$ implies $V_p(v) = 1$ for any $p \in \mathbb{P}$.

As in Example 7.3.8, we can generate a class of interpretation structures $\mathcal{M}_{\mathcal{K}_{int}}$ over Σ_{int} from $\mathcal{K}_{int}$ which preserves the original entailments. Thus, for each $K \in \mathcal{K}_{int}$ consider the interpretation structure

$$M_K = \langle \mathbf{Set}, W, \cdot_{M_K} \rangle$$

over Σ_{int} defined as in Example 7.3.8, but with the following modifications:

- $\neg_{M_K\tau} = \lambda h.\,(\lambda u\vec{a}.\ \bigwedge_{v \in W:\ u \leq v} h(v,\vec{a})^{\mathrm{c}});$
- $\Rightarrow_{M_K\tau} = \lambda h.\,(\lambda u\vec{a}.\ \bigwedge_{v \in W:\ u \leq v} (p_1(h(v,\vec{a})) \Rrightarrow p_2(h(v,\vec{a})))).$

It is easy to prove an analogous of Proposition 7.3.9 for modal intuitionistic propositional logic. Details are left as exercises to the reader. ∇

Example 7.3.11 We now consider first-order logic. Let Σ be a signature for first-order predicate logic as described in Example 7.1.5. Assume we are given a class $\mathbb{I}$ of first-order structures for Σ of the form $I = \langle D, \cdot_I \rangle$. This class defines the global and local entailments as usual. Very briefly, recall that $[\![\varphi]\!]^I \subseteq D^X$ is the set of assignments that make φ true and:

- $\Gamma \vDash^g_{\mathbb{I}} \varphi$ if, for every $I \in \mathbb{I}$, $\bigcap_{\gamma \in \Gamma} [\![\gamma]\!]^I = D^X$ implies $[\![\varphi]\!]^I = D^X$;
- $\Gamma \vDash^\ell_{\mathbb{I}} \varphi$ if, for every $I \in \mathbb{I}$, $\bigcap_{\gamma \in \Gamma} [\![\gamma]\!]^I \subseteq [\![\varphi]\!]^I$.

As was done in the two examples above, we will generate from $\mathbb{I}$ a class of interpretation structures $\mathcal{M}_{\mathbb{I}}$ over Σ such that the original entailments can be recovered.

For each $I \in \mathbb{I}$, let M_I be the interpretation structure $\langle \mathbf{Set}, 1, \cdot_{M_I} \rangle$ over Σ where, using the same notation as in Example 7.3.8:

- $\mathbf{i}_{M_I} = D$;

- $g_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\, g_I(h(\vec{a})))$ for any $g \in \mathcal{G}_n$;
- $\neg_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,(h(\vec{a}))^{\mathrm{c}})$;
- $\wedge_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,(p_1(h(\vec{a})) \sqcap p_2(h(\vec{a}))))$;
- $\vee_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,(p_1(h(\vec{a})) \sqcup p_2(h(\vec{a}))))$;
- $\Rightarrow_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,(p_1(h(\vec{a})) \Rrightarrow p_2(h(\vec{a}))))$;
- $\pi_{M_I\tau} = \lambda h.\,(\lambda u\vec{a}.\,\pi_I(h(u,\vec{a})))$ for any $\pi \in \mathcal{P}_n$;
- $\exists_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,\bigvee_{d\in D} h(\vec{a},d))$;
- $\forall_{M_I\tau} = \lambda h.\,(\lambda\vec{a}.\,\bigwedge_{d\in D} h(\vec{a},d))$.

Again, it is easy to check that each of these families is natural in the sense of Definition 7.3.1. $\triangledown$

Consider $\mathcal{M}_{\mathbb{I}} = \{M_I : I \in \mathbb{I}\}$ and $\mathcal{I}_{\mathbb{I}} = \langle \Sigma, \mathcal{M}_{\mathbb{I}} \rangle$. Then it is again straightforward to prove the following:

Proposition 7.3.12 *Let $\mathbb{I}$ be a class of first-order structures for Σ. Then:*

- $\Gamma \vDash^g_{\mathbb{I}} \varphi$ *if and only if* $\Gamma \vDash^g_{\mathcal{I}_{\mathbb{I}}} \varphi$;
- $\Gamma \vDash^\ell_{\mathbb{I}} \varphi$ *if and only if* $\Gamma \vDash^\ell_{\mathcal{I}_{\mathbb{I}}} \varphi$.

Remark 7.3.13 Of course it is possible to define a signature for first-order intuitionistic logic in our framework, just by modifying the signature of Example 7.1.5 by taking $\neg$ and $\Rightarrow$ as flexible symbols. Thus, every usual Kripke first-order structure I generates a categorical structure identical to the interpretation structure M_I over Σ of Example 7.3.11, but with the obvious modifications for the interpretation of $\neg$ and $\Rightarrow$, as was done in Example 7.3.10. Clearly some additional details concerning Kripke first-order structures must be taken into account, namely:

$$D = \bigcup_{u\in W} D_u$$

such that $\emptyset \neq D_u \subseteq D_v$ whenever $u \leq v$; $\pi^u_I \subseteq D^n_u$ and $\pi^u_I \subseteq \pi^v_I$ whenever $u \leq v$, for $\pi \in \mathcal{P}_n$.

After this, a result analogous to Proposition 7.3.12 can be obtained. Again, the reader is invited to fill the details of this construction, as well as analyze the impact of defining function symbols (including constants) as rigid or flexible. $\triangledown$

Example 7.3.14 We now consider higher-order intuitionistic logic. We will show in this example how the usual topos semantics for higher-order logic can be adapted to our more general setting. Let Σ_{HOL} be the signature described in Example 7.1.7. We establish the semantics of this logic by endowing it with the class $\mathcal{M}^0_{HOL}$ of all structures of the form

$$M = \langle \mathcal{E}, W, \cdot_M \rangle$$

over Σ such that:

- $\mathbf{app}_{\theta\theta' M\tau} = \mathit{trn}(\mathit{eval}(\theta_M, \theta'_M) \circ \mathit{eval}(\tau_M, (\theta'_M)^{\theta_M} \times \theta_M), \tau_M)$ (where the arrow $\mathit{eval}(B, A) : A^B \times B \to A$ is the evaluation map obtained from the definition of exponential A^B);
- $\approx_{\theta M\tau} = \mathit{trn}(\mathit{char}(\mathit{diag}(\theta_M)) \circ \mathit{eval}(\tau_M, \theta_M \times \theta_M), \tau_M)$ (where the diagonal map $\mathit{diag}(A) : A \to A \times A$ is the monomorphism $(\mathrm{id}_A, \mathrm{id}_A)$);
- $\mathbf{set}_{\theta M\tau} = \mathit{trn}(\mathit{trn}(\mathit{eval}(\tau_M \times \theta_M, \Omega) \circ \mathit{can}, \theta_M), \tau_M)$ (where can is the canonical isomorphism from $(\Omega^{\tau_M \times \theta_M} \times \tau_M) \times \theta_M$ to $\Omega^{\tau_M \times \theta_M} \times (\tau_M \times \theta_M)$).

The following diagrams will help to visualize these definitions.

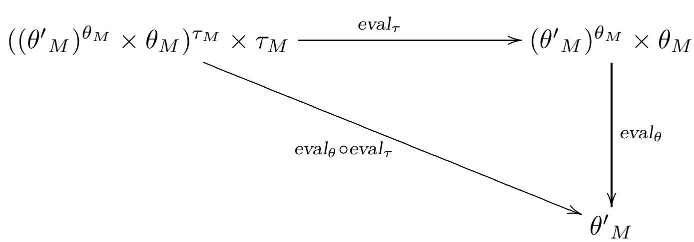


Figure 7.12: Evaluation map

Given the commutative diagram in Figure 7.12, for simplicity we write eval_τ and eval_θ instead of the full names of the respective morphisms, then we define

$$\mathbf{app}_{\theta\theta' M\tau} = \mathit{trn}(\mathit{eval}_\theta \circ \mathit{eval}_\tau, \tau_M) : ((\theta'_M)^{\theta_M} \times \theta_M)^{\tau_M} \to (\theta'_M)^{\tau_M}.$$

Now, given the monomorphism $\mathit{diag}(\theta_M) : \theta_M \to \theta_M \times \theta_M$ defined by

$$\mathit{diag}(\theta_M) = (\mathrm{id}_{\theta_M}, \mathrm{id}_{\theta_M})$$

consider its characteristic map $\mathit{char}(\mathit{diag}(\theta_M)) : \theta_M \times \theta_M \to \Omega$ and the commutative diagram in Figure 7.13.

Then, we define

$$\approx_{\theta M\tau} = \mathit{trn}(\mathit{char}(\mathit{diag}(\theta_M)) \circ \mathit{eval}_\tau, \tau_M) : (\theta_M \times \theta_M)^{\tau_M} \to \Omega^{\tau_M}.$$

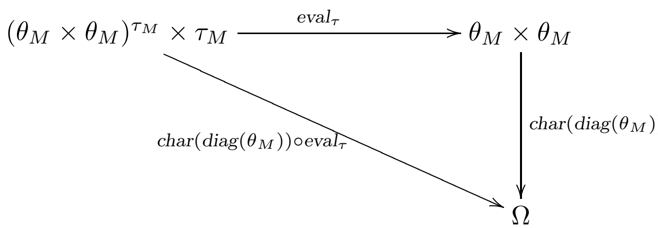


Figure 7.13: Characteristic map

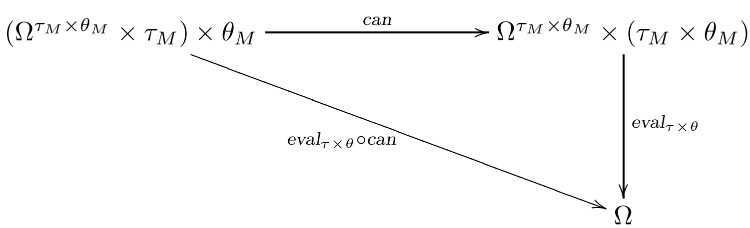


Figure 7.14: Definition of $\mathbf{set}_{\theta M \tau}$

Finally, from the commutative diagram in Figure 7.14 we define, by transposing twice,

$$\mathbf{set}_{\theta M \tau} = trn(trn(eval_{\tau \times \theta} \circ can, \theta_M), \tau_M) : \Omega^{\tau_M \times \theta_M} \to (\Omega^{\theta_M})^{\tau_M}.$$

Clearly, the adaptation of the usual categorical semantics we propose here leaves the entailments unchanged since the extra W has extent 1 (see Proposition 7.3.7) and we have no flexible symbols. Observe that these morphisms are natural by construction. ▽

7.4 Higher-order logic systems

As it was done in the previous chapters, we can put together an interpretation system (recall Definition 7.3.5) and a Hilbert calculus (recall Definition 7.2.6) over the same signature, obtaining a logic system over that signature.

Definition 7.4.1 A *higher-order logic system* is a tuple $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ provided that $\mathcal{I}_\mathcal{L} = \langle \Sigma, \mathcal{M} \rangle$ is an interpretation system and $H_\mathcal{L} = \langle \Sigma, R_g, R_\ell \rangle$ is a Hilbert calculus. ▽

In the sequel, and for simplicity, "logic system" will stand for "higher-order logic system".

Given a logic system $\mathcal{L}$ and $\mathrm{o} \in \{g, \ell\}$, we write $\Psi \vDash^{\mathrm{o}\vec{x}}_{\mathcal{L}} \varphi$ for $\Psi \vDash^{\mathrm{o}\vec{x}}_{\mathcal{I}_\mathcal{L}} \varphi$ and $\Gamma \vdash^{\mathrm{o}\vec{x}}_{\mathcal{L}} \delta \lhd \Pi$ for $\Gamma \vdash^{\mathrm{o}\vec{x}}_{H_\mathcal{L}} \delta \lhd \Pi$.

Definition 7.4.2 A logic system $\mathcal{L}$ is said to be:

- *sound* if, for each $\mathrm{o} \in \{g, \ell\}$, any context $\vec{x}$, and finite $\Psi \cup \{\varphi\} \subseteq gL(\Sigma, \vec{x})$, if $\Psi \vdash^{\mathrm{o}\vec{x}}_{\mathcal{L}} \varphi$ then $\Psi \vDash^{\mathrm{o}\vec{x}}_{\mathcal{L}} \varphi$;
- *complete* if, for each $\mathrm{o} \in \{g, \ell\}$ and $\Psi \cup \{\varphi\} \subseteq gL(\Sigma)$, if $\Psi \vDash^{\mathrm{o}}_{\mathcal{L}} \varphi$ then $\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi$. $\triangledown$

Remark 7.4.3 The definition of soundness just introduced deserves some comments. It seems clear that the intended definition of soundness of a logic system $\mathcal{L}$ should be

$$\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi \text{ implies } \Psi \vDash^{\mathrm{o}}_{\mathcal{L}} \varphi,$$

for $\mathrm{o} \in \{g, \ell\}$. Unfortunately, this definition does not work because of the (possibly) empty domains interpreting the types, which is allowed in categorical semantics. More specifically, from $\Psi, \psi \vDash^{\mathrm{o}}_{\mathcal{L}} \varphi$ and $\Psi \vDash^{\mathrm{o}}_{\mathcal{L}} \psi$ we cannot infer, in general, $\Psi \vDash^{\mathrm{o}}_{\mathcal{L}} \varphi$, for $\mathrm{o} \in \{g, \ell\}$, in case there exists some free variable in ψ not occurring free in $\Psi \cup \{\varphi\}$ (see, for instance, [18]). On the other hand, from $\Psi, \psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi$ and $\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \psi$ we always infer $\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi$ (for $\mathrm{o} \in \{g, \ell\}$) in any logic system $\mathcal{L}$, by the very definition of global and local derivation. This forces us to modify the usual definition of soundness, arriving to the notion introduced in Definition 7.4.2. Note that it is possible to have

$$\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi \text{ but } \Psi \nvDash^{\mathrm{o}}_{\mathcal{L}} \varphi$$

even in a sound logic system $\mathcal{L}$. $\triangledown$

We now specify when a semantic structure is adequate for a Hilbert calculus, a central concept that will be used in the sequel.

Definition 7.4.4 An interpretation structure M over Σ is said to be *appropriate* for a Hilbert calculus H if, for $\mathrm{o} \in \{g, \ell\}$,

$$\Psi \vdash^{\mathrm{o}\vec{x}}_{H} \varphi \text{ implies } \Psi \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma, \{M\}\rangle} \varphi.$$

For each $\mathrm{o} \in \{g, \ell\}$, an interpretation structure M over Σ is said to be *o-appropriate* for a rule $\langle \Gamma, \delta, \Pi\rangle$ over Σ if for every ground substitution ρ over Σ such that $\Pi(\hat{!}_\Sigma \circ \rho) = 1$,

$$\rho(\Gamma) \vDash^{\mathrm{o}}_{\langle \Sigma, \{M\}\rangle} \rho(\delta).$$

$\triangledown$

The reader should note that an interpretation structure over Σ is appropriate for a Hilbert calculus $H = \langle \Sigma, R_g, R_\ell \rangle$ if and only if it is ℓ-appropriate for every local rule in R_ℓ and g-appropriate for every global rule in R_g.

For each $\mathrm{o} \in \{g, \ell\}$, given a set R of rules over Σ, we denote by $Ap_{\mathrm{o}}(R)$ the class of all interpretation structures over Σ that are o-appropriate for the rules in R. Moreover, given a Hilbert calculus H, we define $Ap(H) = Ap_g(R_g) \cap Ap_\ell(R_\ell)$.

Clearly, a logic system $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ is sound if and only if $\mathcal{M} \subseteq Ap(H_{\mathcal{L}})$.

Next definition is the higher-order version of Definition 3.3.11 (propositional based case) and Definition 6.4.8 (first-order based case).

Definition 7.4.5 A logic system $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ is said to be *full* if $\mathcal{M} = Ap(H_{\mathcal{L}})$. $\triangledown$

Example 7.4.6 We consider again modal propositional logic. By endowing the Hilbert calculus of Example 7.2.10 with the class of all interpretation structures appropriate for it we obtain a full logic system that we call $MPL_{\mathbf{K}}$. $\triangledown$

Example 7.4.7 We consider higher-order intuitionistic logic. By endowing the Hilbert calculus of Example 7.2.11 with the class $\mathcal{M}$ of all appropriate interpretation structures we obtain a full logic system that we call *HOL*. $\triangledown$

It is worth noting that the logic system *HOL* contains all the traditional models $\mathcal{M}^0_{HOL}$ described in Example 7.3.14. Indeed:

Proposition 7.4.8 $\mathcal{M}^0_{HOL} \subseteq Ap(H_{HOL})$.

Proof. Let $M \in \mathcal{M}^0_{HOL}$. We need to prove that M is o-appropriate for every rule in R_{o} of *HOL* for $\mathrm{o} \in \{g, \ell\}$.

As mentioned in Remark 7.2.12, in [63] it was proven the equivalence between H_{HOL} restricted to the language of local set theory, and the sequent calculus of Bell [18]. Therefore M is ℓ-appropriate for every axiom of R_ℓ of *HOL* other than $\mathbf{fun}_{\theta\theta'}$.

On the other hand, M is also ℓ-appropriate for axiom $\mathbf{fun}_{\theta\theta'}$. In fact, it is well-known that, in any topos, using the universal properties of the subobject classifier Ω, finite limits and exponentials of the form Ω^A, it is possible to construct arbitrary exponentials (see, for instance, [187]). Since exponentials appear as pullbacks of certain diagrams, the ℓ-appropriateness of M for $\mathbf{fun}_{\theta\theta'}$ follows easily.

Finally, it is straightforward to prove the ℓ-appropriateness of M for MP and the g-appropriateness of M for MP and Gen, because the interpretation of the universal quantifier is the usual in categorical semantics. $\triangleleft$

Observe that, as an immediate consequence of Proposition 7.4.8, the Hilbert calculus H_{HOL} is sound with respect to the (slightly generalized) usual topos semantics $\mathcal{M}^0_{HOL}$ described in Example 7.3.14.

7.5 A general completeness theorem

This section is devoted to obtain sufficient conditions to guarantee that a higher-order logic system is complete. This task is part of the general strategy of proving preservation of completeness by fibring introduced in previous chapters. This strategy consists in the following two steps: (i) to obtain sufficient conditions for completeness; (ii) to prove that these conditions are preserved by fibring. The conditions in step (i) are, as usual, related to the existence some connectives with intuitionistic behaviour, together with the requirement of fullness. Herein, we will establish a general completeness theorem about full logic systems containing *HOL* and with the metatheorem of deduction.

Definition 7.5.1 Let H be a Hilbert calculus. We say that:

- H includes *HOL* if the Hilbert calculus H_{HOL} defined in Example 7.2.11 is embedded in H;
- H has the metatheorem of deduction (*MTD*) if it includes *HOL* and the following condition holds: $\Psi, \psi \vdash^{\ell}_{H} \varphi$ if and only if $\Psi \vdash^{\ell}_{H} (\psi \Rightarrow \varphi)$. ∇

As a consequence of Proposition 2.3.12 we prove the following interesting property of logic systems containing *HOL*:

Proposition 7.5.2 *Let $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ be a logic system including HOL. Then $\mathcal{L}$ has MTD if and only if*

$$\{(\xi \Rightarrow \gamma_1), \ldots, (\xi \Rightarrow \gamma_k)\} \vdash^{\ell}_{\mathcal{L}} (\xi \Rightarrow \delta) \triangleleft \Pi$$

for every local rule $\langle \{\gamma_1, \ldots, \gamma_k\}, \delta, \Pi \rangle \in R_\ell$ and every schema variable $\xi \in \Xi_{\mathbf{\Omega}}$ not occurring in the rule.

In the sequel, we use the following notation. Let Γ be a finite subset of $L(\Sigma)$. Then $(\bigwedge \Gamma)$ denotes a formula obtained from Γ by taking the conjunction of all the formulas in Γ in an arbitrary order and parenthesis association (if $\Gamma = \emptyset$ then we take $(\bigwedge \Gamma)$ to be $\mathbf{t}$). It is easy to prove that, if $\Psi \cup \{\psi\}$ is a finite subset of $gL(\Sigma)$ and $(\bigwedge \Psi)_1$, $(\bigwedge \Psi)_2$ are two conjunctions defined as above then

$$(\bigwedge \Psi)_1 \vdash^{\ell}_{H} (\bigwedge \Psi)_2$$

therefore

$$((\bigwedge \Psi)_1 \Rightarrow \psi) \vdash^{\ell}_{H} ((\bigwedge \Psi)_2 \Rightarrow \psi)$$

in any Hilbert calculus H with *MTD*. Let $\vec{x} = x_1 \ldots x_n$ be a context. Then $(\vec{x} \approx \vec{x})$ denotes a formula of the form $(\bigwedge \{(x_1 \approx x_1), \ldots, (x_n \approx x_n)\})$.

In the proof of the next lemma, and in the rest of this chapter, we adopt the set-theoretic abbreviations in the context of higher-order logic used in [18]. Some of these abbreviations were already used in Example 7.2.11. For instance, U_θ stands for $\{x^\theta \ : \ \mathbf{t}\}$ and $t_2^{t_1}$ for the term

$$\{h \subseteq t_1 \times t_2 \ : \ (\forall x((x \in t_1) \Rightarrow (\exists! y((y \in t_2) \wedge (\langle x, y\rangle \in h))))\}.$$

Lemma 7.5.3 *Every Hilbert calculus H including HOL and with MTD has a canonical model $M_H = \langle \mathcal{E}_H, W_H, \cdot_H\rangle$.*

Proof. Due to its length, this proof (which originally appeared in [62]) is divided into six parts.

Part I: Construction of the linguistic topos $\mathcal{E}_H$.
This first part of the proof parallels (with small adaptations) the standard construction in categorical logic (see for instance [18]). Let $cT(\Sigma)_\theta$ be the set of closed ground Σ-terms of type $\theta \in \Theta(\mathbf{S})$, and consider the collection

$$\mathfrak{A}_\Sigma = \bigcup_{\theta \in \Theta(\mathbf{S})} cT(\Sigma)_{(\theta \to \mathbf{\Omega})}$$

of all the closed ground Σ-terms. Next, we define in $\mathfrak{A}_\Sigma$ the following relation:

$$t_1 \cong_H t_2 \ \text{ if } \ \vdash^\ell_H (t_1 \approx t_2).$$

Clearly, if $t_1 \cong_H t_2$ then $t_1, t_2 \in cT(\Sigma)_{(\theta \to \mathbf{\Omega})}$ for some type θ. It is immediate to see that $\cong_H$ is an equivalence relation. The equivalence class of $t \in \mathfrak{A}_\Sigma$ will be denoted by $[t]_H$ or simply $[t]$.

We define the category $\mathcal{E}_H$ as follows: The objects of $\mathcal{E}_H$ are equivalence classes $[t]$. We use the letters A, B, C, etc. to denote the objects of $\mathcal{E}_H$. Given $A = [t_1]$ and $B = [t_2]$, a morphism in $\mathcal{E}_H$ from A to B is defined to be an equivalence class $[t]$ such that $\vdash^\ell_H (t \in t_2^{t_1})$. It is easy to prove that the notion of morphism is well-defined. If g is a morphism from A to B then we will write, as usual, $g : A \to B$. Given $[t] : [t_1] \to [t_2]$ and $[t'] : [t_2] \to [t_3]$ then the composition map $[t'] \circ [t]$ in $\mathcal{E}_H$ is defined as

$$[\{\langle x, z\rangle \ : (x \in t_1) \wedge (z \in t_3) \wedge (\exists y((y \in t_2) \wedge (\langle x, y\rangle \in t) \wedge (\langle y, z\rangle \in t')))\}].$$

It is immediate to see that $[t'] \circ [t]$ is well-defined and $[t'] \circ [t] : [t_1] \to [t_3]$ is a morphism. Clearly the composition $\circ$ is associative. Given $t \in \mathfrak{A}_\Sigma$ consider the equivalence class

$$\mathrm{id}_{[t]} = [\{\langle x, x\rangle \ : \ x \in t\}].$$

Then, $\mathrm{id}_{[t]} : [t] \to [t]$ is a morphism and $\mathrm{id}_{[t]} \circ g = g$ and $h \circ \mathrm{id}_{[t]} = h$ for $g : A \to [t]$ and $h : [t] \to B$. This shows that, by defining $\mathrm{id}_{[t]}$ as the identity map over the object $[t]$, $\mathcal{E}_H$ is indeed a category.

If $x_i \in X_{\theta_i}$ $(i = 1, \ldots, n)$ such that $x_i \neq x_j$ for $i \neq j$ then $\bar{x}$ will denote the term $\langle x_1, \ldots, x_n\rangle$. Let $t, t' \in \mathfrak{A}_\Sigma$ such that t and t' have type $((\theta_1 \times \cdots \times \theta_n) \to \mathbf{\Omega})$

and $(\theta \to \mathbf{\Omega})$, respectively, and let $\delta \in gT(\Sigma, \vec{x})_\theta$. If $\bar{x} \in t \vdash^{\ell}_{H} \delta \in t'$ then it is immediate to show that the equivalence class

$$[\{\langle \bar{x}, \delta\rangle \ : \ \bar{x} \in t\}]$$

is a morphism from $[t]$ to $[t']$. This morphism which will be denoted by

$$(\bar{x} \mapsto \delta).$$

Observe that this notation (taken from [18]) is somewhat inaccurate. A better notation would be $(\bar{x} \in t \mapsto \delta \in t')$. However, for the sake of simplicity, we prefer to maintain the original notation.

If δ_i is free for y_i in δ (for $1 \leq i \leq m$) then it is not hard to prove that

$$(\bar{y} \mapsto \delta) \circ (\bar{x} \mapsto \langle \delta_1, \dots, \delta_m\rangle) = (\bar{x} \mapsto \delta^{y_1 \dots y_m}_{\delta_1 \dots \delta_m}).$$

To finish this part of the proof, we will give now the sketch of the proof that $\mathcal{E}_H$ is a topos.

Consider 1_H as being the object $[U_{\mathbf{1}}]$. For any object A there exists an unique morphism from A to 1_H given by $(x \mapsto \langle\rangle)$, therefore 1_H is terminal in $\mathcal{E}_H$.

It is easy to prove that, given morphisms $[h_1] : [t_1] \to [t_3]$ and $[h_2] : [t_2] \to [t_3]$, their pullback is given by

$$[\{\langle x, y\rangle : (x \in t_1) \wedge (y \in t_2) \wedge (\exists z((\langle x, z\rangle \in h_1) \wedge (\langle y, z\rangle \in h_2))\}]$$

with the obvious canonical projections.

Consider Ω_H as being the object $[U_{\mathbf{\Omega}}]$ and let $\mathit{true} : 1_H \to \Omega_H$ be the morphism given by $(z^{\mathbf{1}} \mapsto \mathbf{t})$. Then it can be proved that

$$\langle \Omega_H, \mathit{true}\rangle$$

is the subobject classifier in $\mathcal{E}_H$. Given a monomorphism $[t] : A \to B$ in $\mathcal{E}_H$ then its characteristic map $\mathit{char}([t]) : B \to \Omega_H$ is given by

$$(y \mapsto \exists x(\langle x, y\rangle \in t)).$$

Finally, let $[t_1]$ and $[t_2]$ be two objects in $\mathcal{E}_H$. Then it is not hard to prove that the exponential $[t_2]^{[t_1]}$ in $\mathcal{E}_H$ is given by $[t_2^{t_1}]$. The morphism $\mathit{eval}([t_1], [t_2]) : [t_2^{t_1}] \times [t_1] \to [t_2]$ is then given by

$$[\{\langle\langle h, x\rangle, y\rangle \ : \ (h \in t_2^{t_1}) \wedge (x \in t_1) \wedge (\langle x, y\rangle \in h)\}].$$

Moreover, if $[t] : A \times [t_1] \to [t_2]$ then $\mathit{trn}([t], [t_1]) : A \to [t_2^{t_1}]$ is given by

$$(x \mapsto \{\langle y, z\rangle \ : \ \langle\langle x, y\rangle, z\rangle \in t\}).$$

And if $[t] : A \to [t_2^{t_1}]$ then $\mathit{ctr}([t], [t_1]) : A \times [t_1] \to [t_2]$ is given by

$$[\{\langle\langle x, y\rangle, z\rangle \ : \ (x \in A) \wedge (y \in t_1) \wedge (\forall u((\langle x, u\rangle \in t) \ \Rightarrow \ (\langle y, z\rangle \in u)))\}].$$

It is easy to show that $[U_{\theta'}]^{[U_\theta]}$ is isomorphic to $[U_{(\theta\to\theta')}]$ and $eval([U_\theta],[U_{\theta'}])$ is $(\langle h,x\rangle \mapsto \mathbf{app}\langle h,x\rangle)$ in this case. On the other hand, $[U_\theta]\times[U_{\theta'}] = [U_{(\theta\times\theta')}]$.

Part II: The interpretation structure $M_H = \langle \mathcal{E}_H, W_H, \cdot_H\rangle$ over Σ.
Now we define an interpretation structure $M_H = \langle \mathcal{E}_H, W_H, \cdot_H\rangle$ over the linguistic topos $\mathcal{E}_H$ as follows: W_H is 1_H and

- θ_{M_H} is $[U_\theta]$ for every $\theta \in \Theta(\mathbf{S})$;
- $r_{M_H\tau}$ is $(h \mapsto \{\langle \bar{x}, (ry)\rangle \ : \ \langle \bar{x}, y\rangle \in h\})$ whenever $r \in \mathsf{R}_{\theta\theta'}$, $r \neq \mathbf{app}_{\theta''\theta'}$, $r \neq \approx_{\theta'''}$;
- $f_{M_H\tau}$ is $(h \mapsto \{\langle\langle z^{\mathbf{1}}, \bar{x}\rangle, (fy)\rangle \ : \ \langle\langle z^{\mathbf{1}}, \bar{x}\rangle, y\rangle \in h\})$ whenever $f \in \mathsf{F}_{\theta\theta'}$;
- $q_{M_H\tau}$ is $(h \mapsto \{\langle \bar{x}, (qxu)\rangle \ : \ \langle\langle \bar{x}, x\rangle, u\rangle \in h\})$ whenever $q \in \mathsf{Q}_{\theta\theta'\theta''}$, $q \neq \mathbf{set}_\theta$.

The interpretation in M_H of $\mathbf{app}_{\theta\theta'}$, $\approx_\theta$ and $\mathbf{set}_\theta$ is as in Example 7.3.14. In order to better understand its definition, we proceed now to briefly analyze them. Considering $[U_{\theta'}]^{[U_\theta]}$ as $[U_{(\theta\to\theta')}]$ (they are isomorphic, as pointed above) consider the morphism

$$g = eval([U_\theta],[U_{\theta'}]) \circ eval([U_\tau],[U_{(\theta\to\theta')}]\times[U_\theta]).$$

From the results proved in Part I we obtain that

$$g = (\langle h',x\rangle \mapsto \mathbf{app}\langle h',x\rangle) \circ [\{\langle\langle h,\bar{x}\rangle, u\rangle \ : \ \langle \bar{x}, u\rangle \in h\}].$$

Thus, it is immediate that

$$g = [\{\langle\langle h,\bar{x}\rangle, \mathbf{app}\, u\rangle \ : \ \langle \bar{x}, u\rangle \in h\}]$$

and so

$$\mathbf{app}_{\theta\theta' M_H\tau} = trn(g,[U_\tau]) = (h \mapsto \{\langle \bar{x}, \mathbf{app}\, u\rangle \ : \ \langle \bar{x}, u\rangle \in h\}).$$

With respect to the equality symbols, consider the diagonal map

$$diag([U_\theta]) = (x \mapsto \langle x,x\rangle) : [U_\theta] \to [U_\theta]\times[U_\theta].$$

Using again Part I and the deduction rules of *HOL* we obtain

$$char(diag([U_\theta])) = (y \mapsto \exists x(\langle x,y\rangle \in diag([U_\theta]))) = (y \mapsto ((y)_1 \approx (y)_2)).$$

Consider the morphism

$$g = char(diag([U_\theta])) \circ eval([U_\tau],[U_\theta]\times[U_\theta]).$$

Then $g = [\{\langle\langle h,\bar{x}\rangle, ((y)_1 \approx (y)_2)\rangle \ : \ \langle \bar{x}, y\rangle \in h\}]$, therefore

$$\approx_{\theta M_H\tau} = trn(g,[U_\tau]) = (h \mapsto \{\langle \bar{x}, ((y)_1 \approx (y)_2)\rangle \ : \ \langle \bar{x}, y\rangle \in h\}).$$

Finally, consider $[U_{\mathbf{\Omega}}]^{[U_\theta]}$ as $[U_{(\theta\to\mathbf{\Omega})}]$ (they are isomorphic) and define

$$g = eval([U_\tau]\times[U_\theta],[U_{\mathbf{\Omega}}])\circ can$$

where

$$can : ([U_{\mathbf{\Omega}}]^{[U_\tau]\times[U_\theta]}\times[U_\tau])\times[U_\theta]\to[U_{\mathbf{\Omega}}]^{[U_\tau]\times[U_\theta]}\times([U_\tau]\times[U_\theta])$$

is the canonical isomorphism. Using the results in Part I and the rules of *HOL* it is easy to prove that

$$\begin{aligned} g &= [\{\langle\langle h',\langle\bar{x},x\rangle\rangle,v\rangle \;:\langle\langle\bar{x},x\rangle,v\rangle\in h'\}]\circ[\{\langle\langle\langle h,\bar{x}\rangle,x\rangle,\langle h,\langle\bar{x},x\rangle\rangle\rangle \;:\; \mathbf{t}\}]\\ &= [\{\langle\langle\langle h,\bar{x}\rangle,x\rangle,v\rangle \;:\langle\langle\bar{x},x\rangle,v\rangle\in h\}].\end{aligned}$$

If we define $g_1 = trn(g,[U_\theta])$ then it is immediate that

$$g_1 = [\{\langle\langle h,\bar{x}\rangle,u\rangle \;:\; (\forall x(\forall v(((x\in u)\approx v) \;\Leftrightarrow\; \langle\langle\bar{x},x\rangle,v\rangle\in h)))\}].$$

Thus, the morphism $\mathbf{set}_{\theta M_H\tau} = trn(g_1,[U\tau])$ is given by

$$\mathbf{set}_{\theta M_H\tau} = (h\mapsto\{\langle\bar{x},u\rangle \;:\; (\forall x(\forall v(((x\in u)\approx v) \;\Leftrightarrow\; \langle\langle\bar{x},x\rangle,v\rangle\in h)))\}).$$

In order to prove that M_H is indeed an interpretation structure over Σ it remains to prove the naturality of the morphisms $r_{M_H\tau}$ (where $r\neq\mathbf{app}_{\theta''\theta'}$ and $r\neq\approx_{\theta'''}$), $f_{M_H\tau}$ and $q_{M_H\tau}$ (where $q\neq\mathbf{set}_\theta$). In fact, the families of morphisms $\mathbf{app}_{\theta''\theta'}$, $\approx_{\theta'''}$ and $\mathbf{set}_\theta$ are natural by construction, as pointed in Example 7.3.14. We will only prove the naturality of the family of morphisms $r_{M_H\tau}$, because the proof for the other cases is similar. Since $W_H = 1_H$ then

$$r_{M_H\tau} = (h\mapsto\{\langle\bar{x},(ry)\rangle : \langle\bar{x},y\rangle\in h\})$$

for every τ. Now, consider morphisms $m=[t]:U_\tau\to U_{\tau'}$ and $n=[t']:U_{\tau'}\to U_\theta$. Then

$$r_{M\tau'}\circ trn(n,\tau'_M) = (z^{\mathbf{1}}\mapsto\{\langle\bar{x},(ry)\rangle : \langle\bar{x},y\rangle\in t'\}).$$

From the results stated in Part I and the rules of *HOL* we easily obtain that

$$ctr(r_{M\tau'}\circ trn(n,\tau'_M),\tau'_M)\circ m = [\{\langle\bar{y},(ry)\rangle : (\exists\bar{x}(\langle\bar{y},\bar{(}x\rangle\in t)\wedge(\langle\bar{x},y\rangle\in t')))\}].$$

On the other hand, it is immediate to prove that

$$r_{M\tau}\circ trn(n\circ m,\tau_M) = (z^{\mathbf{1}}\mapsto\{\langle\bar{y},(ry)\rangle : (\exists\bar{x}((\langle\bar{y},\bar{x}\rangle\in t)\wedge(\langle\bar{x},y\rangle\in t')))\}).$$

Hence we obtain the desired naturality of the family of morphisms $r_{M_H\tau}$:

$$ctr(r_{M\tau'}\circ trn(n,\tau'_M),\tau'_M)\circ m = ctr(r_{M\tau}\circ trn(n\circ m,\tau_M),\tau_M).$$

Part III: $[\![t]\!]_{\vec{x}}^{M_H} = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto t)$.
Let $t \in gT(\Sigma, \vec{x})_\theta$. Since W_H is 1_H (therefore $[U_\theta] \simeq W_H \times [U_\theta]$ for all θ) we prove now by induction on the complexity of t that

$$[\![t]\!]_{\vec{x}}^{M_H} = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto t) : 1_H \times [U_\tau] \to [U_\theta]$$

where $z^{\mathbf{1}} \in X_{\mathbf{1}}$ does not occur in $\vec{x}$. If t is a variable or t is $\langle\rangle$ the result is immediate. If t is $\langle t_1, \ldots, t_n\rangle$ or t is $(t')_i$ the conclusion follows easily by induction hypothesis. If t is $\mathbf{app}_{\theta\theta'}\langle t_1, t_2\rangle$ then, using the induction hypothesis, $[\![\langle t_1, t_2\rangle]\!]_{\vec{x}}^{M_H} = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \langle t_1, t_2\rangle)$. Consider the morphism

$$g = \mathbf{app}_{\theta\theta' M_H \tau} \circ trn([\![\langle t_1, t_2\rangle]\!]_{\vec{x}}^{M_H}, [U_\tau]).$$

Using the results in Parts I and II we obtain that

$$\begin{aligned} g &= (h \mapsto \{\langle \bar{x}, \mathbf{app}\, u\rangle \ : \ \langle \bar{x}, u\rangle \in h\}) \circ (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, \langle t_1, t_2\rangle\rangle \ : \ \mathbf{t}\}) \\ &= (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, \mathbf{app}\langle t_1, t_2\rangle\rangle \ : \ \mathbf{t}\}). \end{aligned}$$

From this it is clear that

$$[\![\mathbf{app}\langle t_1, t_2\rangle]\!]_{\vec{x}}^{M_H} = ctr(g, [U_\tau]) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \mathbf{app}\langle t_1, t_2\rangle).$$

Now, if t is $(t_1 \approx_\theta t_2)$ let g be the morphism $\approx_{\theta M_H \tau} \circ\, trn([\![\langle t_1, t_2\rangle]\!]_{\vec{x}}^{M_H}, [U_\tau])$. Then g is given by

$$(h \mapsto \{\langle \bar{x}, ((y)_1 \approx_\theta (y)_2)\rangle \ : \ \langle \bar{x}, y\rangle \in h\}) \circ (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, \langle t_1, t_2\rangle\rangle \ : \ \mathbf{t}\})$$

and so $g = (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, (t_1 \approx_\theta t_2)\rangle \ : \ \mathbf{t}\})$. From this follows easily that

$$[\![(t_1 \approx_\theta t_2)]\!]_{\vec{x}}^{M_H} = ctr(g, [U_\tau]) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto (t_1 \approx_\theta t_2)).$$

Now, if t is the ground term $\mathbf{set}_\theta x\varphi$ let y be a variable free for x in φ not occurring in $\vec{x}$. By induction hypothesis,

$$[\![\varphi_y^x]\!]_{\vec{x}y}^{M_H} = (\langle z^{\mathbf{1}}, \langle \bar{x}, y\rangle\rangle \mapsto \varphi_y^x).$$

Let $g = \mathbf{set}_{\theta M_H \tau} \circ trn([\![\varphi_y^x]\!]_{\vec{x}y}^{M_H}, [U_\tau] \times [U_\theta])$. By Part II it follows that $\mathbf{set}_{\theta M_H \tau}$ is

$$(h \mapsto \{\langle \bar{x}, u\rangle \ : \ (\forall x(\forall v(((x \in u) \approx v) \ \Leftrightarrow \ (\langle\langle \bar{x}, x\rangle, v\rangle \in h))))\}).$$

On the other hand, by Part I we obtain that $trn([\![\varphi_y^x]\!]_{\vec{x}y}^{M_H}, [U_\tau] \times [U_\theta])$ is

$$(z^{\mathbf{1}} \mapsto \{\langle\langle \bar{x}, y\rangle, \varphi_y^x\rangle \ : \ \mathbf{t}\}).$$

Thus, using the deduction rules of *HOL*, it follows that

$$\begin{aligned} g &= (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, u\rangle \;:\; (\forall x(\forall v(((x \in u) \approx v) \;\Leftrightarrow\; (v \approx \varphi))))\}) \\ &= (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, u\rangle \;:\; (\forall x((x \in u) \;\Leftrightarrow\; \varphi))\}) \\ &= (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, \{x \;:\; \varphi\}\rangle \;:\; \mathbf{t}\}). \end{aligned}$$

From this follows easily that

$$[\![\mathbf{set}_\theta x\varphi]\!]_{\vec{x}}^{M_H} = [\![\{x \;:\; \varphi\}]\!]_{\vec{x}}^{M_H} = ctr(g, [U_\tau]) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \{x \;:\; \varphi\}).$$

If t is (ft'), with $f \in \mathsf{F}_{\theta\theta'}$ and $t \in gT(\Sigma)_\theta$ then, by induction hypothesis,

$$[\![t']\!]_{\vec{x}}^{M_H} = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto t').$$

Let $g = f_{M_H\tau} \circ trn([\![t']\!]_{\vec{x}}^{M_H}, W_H \times [U_\tau])$ and let $x^{\mathbf{1}} \in X_{\mathbf{1}}$ not occurring in $\langle z^{\mathbf{1}}, \bar{x}\rangle$. Since $f_{M_H\tau}$ is, by definition,

$$(h \mapsto \{\langle\langle z^{\mathbf{1}}, \bar{x}\rangle, (fy)\rangle \;:\; \langle\langle z^{\mathbf{1}}, \bar{x}\rangle, y\rangle \in h\})$$

and $trn([\![t']\!]_{\vec{x}}^{M_H}, W_H \times [U_\tau])$ is the morphism

$$(x^{\mathbf{1}} \mapsto \{\langle\langle z^{\mathbf{1}}, \bar{x}\rangle, t'\rangle \;:\; \mathbf{t}\})$$

then, using the rules of *HOL*, it is straightforward to prove that

$$g = (x^{\mathbf{1}} \mapsto \{\langle\langle z^{\mathbf{1}}, \bar{x}\rangle, (ft')\rangle \;:\; \mathbf{t}\}).$$

Therefore

$$[\![(ft')]\!]_{\vec{x}}^{M_H} = ctr(g, W_H \times [U_\tau]) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto (ft'))$$

as required. The proof for the case $t = (rt')$ with $r \in \mathsf{R}_{\theta\theta'}$ is analogous. Finally, suppose that t is qxt', where $q \in \mathsf{Q}_{\theta\theta'\theta''}$, $x \in X_\theta$ and $t' \in gT(\Sigma)_{\theta'}$. Consider a variable y of type θ free for x in t' not occurring in $\vec{x}$. By induction hypothesis,

$$[\![{t'}^x_y]\!]_{\vec{x}y}^{M_H} = (\langle z^{\mathbf{1}}, \langle \bar{x}, y\rangle\rangle \mapsto {t'}^x_y).$$

Let $g = q_{M_H\tau} \circ trn([\![{t'}^x_y]\!]_{\vec{x}y}^{M_H}, [U_\tau] \times [U_\theta])$. Since $q_{M_H\tau}$ is the morphism

$$(h \mapsto \{\langle \bar{x}, (qxu)\rangle \;:\; \langle\langle \bar{x}, x\rangle, u\rangle \in h\}),$$

by definition, and $trn([\![{t'}^x_y]\!]_{\vec{x}y}^{M_H}, [U_\tau] \times [U_\theta])$ is

$$(z^{\mathbf{1}} \mapsto \{\langle\langle \bar{x}, y\rangle, {t'}^x_y\rangle \;:\; \mathbf{t}\})$$

then, using again the deduction rules of *HOL*, we obtain that

$$g = (z^{\mathbf{1}} \mapsto \{\langle \bar{x}, (qxt')\rangle \;:\; \mathbf{t}\}).$$

Then

$$[\![(qxt')]\!]_{\vec{x}}^{M_H} = ctr(g, [U_\tau]) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto (qxt')).$$

Part IV: $\Psi \vdash_H^\ell \varphi$ implies $\Psi \vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$.
Consider $\varphi \in gL(\Sigma)$ with canonical context $\vec{x}$. Then $\vdash_H^\ell \varphi$ implies $\vdash_H^\ell (\varphi \approx \mathbf{t})$ implies $(\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \varphi) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \mathbf{t})$ implies $[\![\varphi]\!]_{\vec{x}}^{M_H} = [\![\mathbf{t}]\!]_{\vec{x}}^{M_H}$ (by Part III) implies $\vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$.
Consider now a finite set $\Psi \cup \{\varphi\}$ of ground Σ-formulas and let $(\bigwedge \Psi)$ be a formula defined using the notation introduced after Definition 7.5.1. Then $\Psi \vdash_H^\ell \varphi$ implies $\vdash_H^\ell ((\bigwedge \Psi) \Rightarrow \varphi)$, because H has *MTD*. Therefore, $\vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell ((\bigwedge \Psi) \Rightarrow \varphi)$ and so $\Psi \vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$ as desired.

Part V: $\Psi \vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$ implies $\Psi \vdash_H^\ell \varphi$.
In this step we verify that M_H is a canonical model for H (with respect to local derivations). Let $\varphi \in gL(\Sigma)$ with canonical context $\vec{x}$. Then $\vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$ implies $[\![\varphi]\!]_{\vec{x}}^{M_H} = [\![\mathbf{t}]\!]_{\vec{x}}^{M_H}$ implies $(\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \varphi) = (\langle z^{\mathbf{1}}, \bar{x}\rangle \mapsto \mathbf{t})$ (by Part III) implies $\vdash_H^\ell (\varphi \approx \mathbf{t})$ implies $\vdash_H^\ell \varphi$.
Consider now a finite subset $\Psi \cup \{\varphi\}$ of $gL(\Sigma)$. Then we obtain: $\Psi \vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell \varphi$ implies $\vDash_{\langle\Sigma,\{M_H\}\rangle}^\ell ((\bigwedge \Psi) \Rightarrow \varphi)$ implies $\vdash_H^\ell ((\bigwedge \Psi) \Rightarrow \varphi)$ implies $\Psi \vdash_H^\ell \varphi$, because H has *MTD*.

Part VI: M_H is appropriate for H.
By Part IV, it remains to prove appropriateness of M_H with respect to global derivations. Let $\Psi \cup \{\varphi\}$ be a finite subset of $gL(\Sigma)$ such that $\Psi \vdash_H^{g\vec{x}} \varphi$ and

$$\bigwedge_{\psi \in \Psi} [\![\psi]\!]_{\vec{x}}^{M_H} = true_{W \times \theta_{\vec{x} M_H}}.$$

Thus, $\vDash_{\langle\Sigma,\{M_H\}\rangle}^{\ell\vec{x}} \psi$ for every $\psi \in \Psi$, and so $\vDash_{\langle\Sigma,\{M_H\}\rangle}^{\ell\vec{x}} (\psi \wedge (\vec{x} \approx \vec{x}))$. In other words,

$$\vDash_{\langle\Sigma,\{M_H\}\rangle}^{\ell} (\psi \wedge (\vec{x} \approx \vec{x})).$$

By Part V we obtain $\vdash_H^\ell (\psi \wedge (\vec{x} \approx \vec{x}))$ and so $\vdash_H^{\ell\vec{x}} \psi$ for every $\psi \in \Psi$. Thus $\vdash_H^{\ell\vec{x}} \varphi$ and then $\vdash_H^\ell (\varphi \wedge (\vec{x} \approx \vec{x}))$. By Part IV we get

$$\vDash_{\langle\Sigma,\{M_H\}\rangle}^{\ell} (\varphi \wedge (\vec{x} \approx \vec{x})).$$

Therefore, $[\![\varphi]\!]_{\vec{x}}^{M_H} = true_{W \times \theta_{\vec{x} M_H}}$. That is, $\Psi \vDash_{\langle\Sigma,\{M_H\}\rangle}^{g\vec{x}} \varphi$ as desired. $\triangleleft$

It is worth noting that the reduct to Σ_{HOL} of M_H belongs to $\mathcal{M}_{HOL}^0$. This means that M_H is standard with respect to the language of pure *HOL*.

Proposition 7.5.4 *Every full logic system with Hilbert calculus including HOL and with MTD is complete.*

Proof. Let $\mathcal{I} = \langle \Sigma, \mathcal{M} \rangle$, where $\mathcal{M}$ is the class of interpretation structures over Σ appropriate for H, and let $\Psi \cup \{\varphi\}$ be a finite set of ground formulas. Suppose that $\Psi \vDash^{\ell}_{\mathcal{I}} \varphi$. Since M_H is appropriate for H then $\Psi \vDash^{\ell}_{\langle \Sigma, \{M_H\} \rangle} \varphi$. Thus $\Psi \vdash^{\ell}_{H} \varphi$, by Part V of the proof of Lemma 7.5.3.

Now, suppose that $\Psi \vDash^{g}_{\mathcal{I}} \varphi$. Let $\vec{x}$ be the canonical context of $\Psi \cup \{\varphi\}$ and let H^{Ψ} be the Hilbert calculus obtained from H by adding the axiom

$$\langle \emptyset, ((\bigwedge \Psi) \wedge (\vec{x} \approx \vec{x})), \mathbf{U} \rangle$$

where $(\bigwedge \Psi)$ and $(\vec{x} \approx \vec{x})$ are defined using the notation stated after Definition 7.5.1. If $\mathcal{M}^{\Psi}$ is the class of interpretation structures over Σ appropriate for H^{Ψ} then

$$\mathcal{M}^{\Psi} = \{M \in \mathcal{M} : [\![(\bigwedge \Psi) \wedge (\vec{x} \approx \vec{x})]\!]^{M}_{\vec{x}} = true_{W \times \theta_{\vec{x}M}}\}.$$

Let $\mathcal{I}^{\Psi} = \langle \Sigma, \mathcal{M}^{\Psi} \rangle$. Since $\Psi \vDash^{g}_{\mathcal{I}} \varphi$ then $\vDash^{g}_{\mathcal{I}^{\Psi}} (\varphi \wedge (\vec{x} \approx \vec{x}))$ and so, by Definition 7.2.9, $\vDash^{\ell}_{\mathcal{I}^{\Psi}} (\varphi \wedge (\vec{x} \approx \vec{x}))$. By Part VI of the proof of Lemma 7.5.3 we have that

$$\vDash^{\ell}_{\langle \Sigma, \{M_{\mathcal{D}^{\Psi}}\} \rangle} (\varphi \wedge (\vec{x} \approx \vec{x})).$$

Therefore $\vdash^{\ell}_{H^{\Psi}} (\varphi \wedge (\vec{x} \approx \vec{x}))$, by Part V of the proof of the same lemma. Hence $\Psi \vdash^{g}_{H} \varphi$. ◁

7.6 Fibring higher-order logic systems

This section finally introduces the category of higher-order logic systems by defining a suitable notion of morphism between them. In this category both constrained and unconstrained forms of fibring will be defined as universal constructions. Then it will be shown that soundness is always preserved by fibring. After this, we will address the problem of preservation of completeness by fibring. Firstly, we will obtain a result in the case of rich logics (that is, logics including *HOL* and with the *MTD*). Finally, we will extend this result to weaker logics under the assumption of the preservation of the conservativeness of *HOL*-enrichment.

The following concepts are necessary in order to define logic system morphisms.

Let $h : \Sigma \to \Sigma'$ be a signature morphism. Given a set R of rules over Σ, the *image* of R by h, denoted by $h(R)$, is the set

$$\{\langle h(\Gamma), h(\delta), \Pi \rangle : \langle \Gamma, \delta, \Pi \rangle \in R\}$$

of rules over Σ'. Given a Hilbert calculus $H = \langle \Sigma, R_g, R_\ell \rangle$, let $h(H)$ be the Hilbert calculus $\langle \Sigma', h(R_g), h(R_\ell) \rangle$. And given an interpretation structure $M' = \langle \mathcal{E}', W', \cdot_{M'} \rangle$ over Σ', the *reduct* of M' along h, denoted by $M'|_h$, is the interpretation structure $\langle \mathcal{E}', W', \cdot_{M'} \circ h \rangle$ over Σ.

Definition 7.6.1 Let $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ and $\mathcal{L}' = \langle \Sigma', \mathcal{M}', R'_g, R'_\ell \rangle$ be logic systems. A *logic system morphism* $h : \mathcal{L} \to \mathcal{L}'$ is a signature morphism $h : \Sigma \to \Sigma'$ such that:

1. $M' \in \mathcal{M}'$ implies $M'|_h \in \mathcal{M}$;
2. for every $M' \in \mathcal{M}'$, $M'|_h \in Ap(H_\mathcal{L})$ implies $M' \in Ap(h(H_\mathcal{L}))$;
3. $h(R_g) \subseteq R'_g$;
4. $h(R_\ell) \subseteq R'_\ell$. ∇

From previous work on the subject of fibring (see for instance [282, 240]), conditions (1), (3) and (4) are to be expected. However, (3) and (4) are a bit stronger than the corresponding conditions for propositional Hilbert calculi defined in Chapter 2. It is not hard to prove that, concerning the fibring results to be stated below, nothing changes if we consider this more restricted notion of morphism. With respect to condition (2), it is a reasonable requirement that will allow the preservation of soundness by fibring. This condition should be seen more as a requirement on the rules of $H_\mathcal{L}$ than on the models. Consider, for instance, the usual rule of quantified logic

$$\langle \emptyset, ((\forall x\, \xi_1) \Rightarrow \xi_1{}^{x}_{\xi_2}),\ \xi_2 \rhd x : \xi_1 \rangle$$

with the standard proviso: No free variable in ξ_2 must be captured by a quantifier when x is replaced by ξ_2 in ξ_1. Surprisingly, this rule becomes unsound when put in an environment (signature) where modalities and flexible symbols are available. More precisely, condition (2) will be violated when this rule is present in $\mathcal{L}$ and $\mathcal{L}'$ contains modalities and other flexible symbols.

Suppose, for example, that $\mathcal{L}'$ is a temporal quantified logic. If we substitute ξ_1 by be the formula $((s \approx x) \Rightarrow (\mathbf{F}(s > x)))$, where s is flexible and $\mathbf{F}$ is the "sometime in the future" modality, and if we substitute ξ_2 by the term s, then, $\xi_1{}^{x}_{\xi_2}$ becomes $((s \approx s) \Rightarrow (\mathbf{F}(s > s)))$. Hence we started with the formula $((s \approx x) \Rightarrow (\mathbf{F}(s > x)))$ that can be true in some interpretation structure and get, after the substitution, the formula $((s \approx s) \Rightarrow (\mathbf{F}(s > s)))$ that can never be true. We want to avoid this unpleasant situation.

However, it is possible to transform the rules in order to obtain a more robust version of them, avoiding the kind of problems with condition (2) pointed out before. In the last example, it is enough to reinforce the proviso with the additional requirement that no flexible symbol in ξ_2 falls into the scope of a modality when x is replaced by ξ_2 in ξ_1. Of course this (stronger) proviso changes nothing in the original quantified logic, but it makes all the difference when embedding it into a richer logic, such as the ones generally obtained by fibring.

The last remark suggests the following definition:

Definition 7.6.2 A Hilbert calculus H is said to be *robust* if, for every signature monomorphism $h : \Sigma \to \Sigma'$ and interpretation structure M' over Σ',

$$M'|h \in Ap(H) \ \text{ implies } \ M' \in Ap(h(H)).$$

A logic system $\mathcal{L}$ is said to be robust if $H_{\mathcal{L}}$ is robust. $\triangledown$

Remark 7.6.3 It is worth noting that it is quite easy to make robust any given logic system: The *brute force* method (including in all rules the additional requirement forbidding foreign categories of symbols) always works. For instance, if a logic system has no flexible symbols then we include in the proviso of every rule the additional requirement that it may not be applied when substitution ρ uses flexible symbols. This changes nothing in the original logic system but makes it much weaker when combined with other logic systems. $\triangledown$

Example 7.6.4 We consider higher-order intuitionistic logic. From now on, we assume that the Hilbert calculus of *HOL* (recall Example 7.2.11) is made robust, namely by interpreting the proviso $\xi_2 \rhd x : \xi_1$ as forbidding both (i) capture of free variables in ξ_2 by binding operators in ξ_1, and (ii) capture of flexible symbols in ξ_2 by flexible symbols in ξ_1. $\triangledown$

We can now introduce the category of logic systems.

Definition 7.6.5 The category **HLog** of logic systems is defined as follows: Its objects are logic systems (cf. Definition 7.4.1), and its morphisms are logic systems morphisms (cf. Definition 7.6.1). Identity maps and composition are inherited from the category **HSig** of signatures. $\triangledown$

In the category **HLog**, fibrings are defined through universal constructions, as we shall see below.

Definition 7.6.6 Given two logic systems $\mathcal{L}'$ and $\mathcal{L}''$, their *unconstrained fibring* is the logic system $\mathcal{L} = \mathcal{L}' \oplus \mathcal{L}''$ such that:

- $\Sigma = \Sigma' \oplus \Sigma''$ with injections i' and i'' (coproduct in **HSig** of Σ' and Σ'');
- $\mathcal{M} = \{M \in Str(\Sigma' \oplus \Sigma'') : M|_{i'} \in \mathcal{M}', \ M|_{i''} \in \mathcal{M}'',$
  $M|_{i'} \in Ap(H') \text{ implies } M \in Ap(i'(H')), \text{ and}$
  $M|_{i''} \in Ap(H'') \text{ implies } M \in Ap(i''(H''))\};$
- $R_g = i'(R'_g) \cup i''(R''_g)$;
- $R_\ell = i'(R'_\ell) \cup i''(R''_\ell)$. $\triangledown$

We shall prove that the unconstrained fibring is a coproduct in the category **HLog**. Previous to this, we need to state a useful result.

Lemma 7.6.7 *Let $h : \Sigma \to \Sigma'$ be a signature morphism.*

1. *Let σ be a substitution over Σ and consider the substitution $\sigma' = \hat{h} \circ \sigma$ over Σ'. Then $\hat{\sigma}' \circ \hat{h} = \hat{h} \circ \hat{\sigma}$.*

2. *Let M' be an interpretation structure over Σ', $\rho \in gSbs(\Sigma)$ and $\rho' \in gSbs(\Sigma')$ such that $\rho' = \hat{h} \circ \rho$. Then $[\![\rho(t)]\!]_{\vec{x}}^{M'|_h} = [\![\rho'(\hat{h}(t))]\!]_{\vec{x}}^{M'}$ for every $t \in T(\Sigma, \vec{x})$.*

3. *Let H be a Σ-Hilbert calculus and $M' \in Str(\Sigma')$. Then $M' \in Ap(h(H))$ implies $M'|_h \in Ap(H)$.*

Proof. 1. Given h and σ we must prove that, by defining σ' such that the diagram in Figure 7.15 commutes for every type θ. Then the diagram in Figure 7.16 is commutative for every type θ.

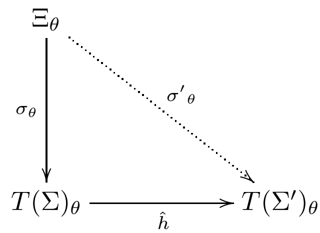


Figure 7.15: Substitution σ' over Σ'

It is easy to prove by induction on the complexity of a term $t \in T(\Sigma, \vec{x})$ (where, by convention, $\xi_{\xi'}^{x}$ has complexity 1) that a variable x occurs free in t if and only if it occurs free in $\hat{h}(t)$. Then, it is immediate to prove that a term t is free for a variable x in a term t' if and only if $\hat{h}(t)$ is free for x in $\hat{h}(t')$. From these facts the result follows by induction on the complexity of the term.

2. Immediate from item (1) and our definitions.

3. Immediate from item (2) and our definitions. ◁

Proposition 7.6.8 *The unconstrained fibring $\mathcal{L}' \oplus \mathcal{L}''$ is the coproduct in* ***HLog*** *of $\mathcal{L}'$ and $\mathcal{L}''$.*

Proof. Let $\mathcal{L}$ be the logic system $\mathcal{L}' \oplus \mathcal{L}'' = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ constructed as in Definition 7.6.6. It is immediate to conclude that the injections $i' : \Sigma' \to \Sigma$ and $i'' : \Sigma'' \to \Sigma$ are morphisms in **HLog**. Consider a logic system

$$\check{\mathcal{L}} = \langle \check{\Sigma}, \check{\mathcal{M}}, \check{R}_g, \check{R}_\ell \rangle$$

and morphisms $j' : \mathcal{L}' \to \check{\mathcal{L}}$, $j'' : \mathcal{L}'' \to \check{\mathcal{L}}$ in **HLog**. Then, there is a unique signature morphism $h : \Sigma \to \check{\Sigma}$ such that $h \circ i' = j'$ and $h \circ i'' = j''$ in **HSig**.

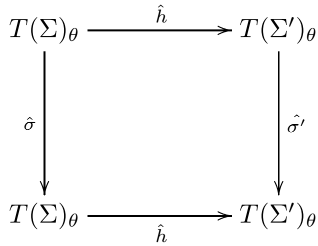


Figure 7.16: $\hat{\sigma}' \circ \hat{h} = \hat{h} \circ \hat{\sigma}$

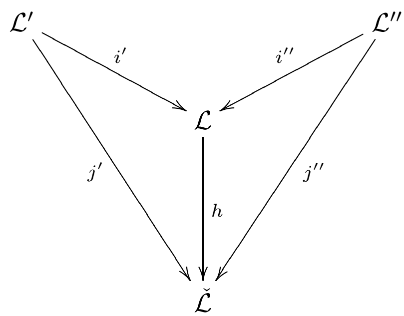


Figure 7.17: Unconstrained fibring

It suffices to show that h is morphism in **HLog**, that is, there is a commutative diagram in **HLog** as showed in Figure 7.17.

Let $\check{M} \in \check{\mathcal{M}}$. Firstly, we need to show that

$$\check{M}|_h \in \mathcal{M}$$

that is, $\check{M}|_h$ satisfies the conditions defining the elements of the class $\mathcal{M}$ (see Definition 7.6.6). Observe that

$$(\check{M}|_h)|_{i'} = \check{M}|_{j'}$$

and then it belongs to $\mathcal{M}'$, because j' is a **HLog**-morphism. Analogously we show that $(\check{M}|_h)|_{i''} \in \mathcal{M}''$. Assume that $(\check{M}|_h)|_{i'}$ belongs to $Ap(H')$. Then

$$\check{M} \in Ap(h(i'(H')))$$

because $j' = h \circ i'$ is a morphism in **HLog**. Using Lemma 7.6.7(3) we infer that $\check{M}|_h \in Ap(i'(H'))$. Analogously we prove that $(\check{M}|_h)|_{i''} \in Ap(H'')$ implies $\check{M}|_h \in Ap(i''(H''))$. From these properties follows that $\check{M}|_h \in \mathcal{M}$ and so h satisfies condition (1) of Definition 7.6.1.

Suppose now that $\check{M} \in \check{\mathcal{M}}$ is such that $\check{M}|_h \in Ap(H)$. In particular, by definition of H, $\check{M}|_h \in Ap(i'(H'))$ and so, since $(\check{M}|_h)|_{i'} = \check{M}|_{j'}$, we obtain that

$$\check{M}|_{j'} \in Ap(H')$$

by Lemma 7.6.7(3). Thus $\check{M} \in Ap(h(i'(H')))$ since $j' = h \circ i'$ is a morphism in **HLog**. Using the fact that $\check{M}|_h \in Ap(i''(H''))$ we prove, by an analogous argument, that

$$\check{M} \in Ap(h(i''(H''))).$$

Thus, $\check{M} \in Ap(h(H))$, by definition of H, and so h satisfies condition (2) of Definition 7.6.1.

Finally, by definition of H and the fact that $j' = h \circ i'$ and $j'' = h \circ i''$ are morphisms in **HLog** we have that $h(R_{\text{o}}) \subseteq \check{R}_{\text{o}}$ for $\text{o} \in \{g, \ell\}$. Thus, h also satisfies conditions (3) and (4) of Definition 7.6.1 and then h is a morphism in **HLog** which commutes the diagram above. The uniqueness of h in **HLog** is inherited from the uniqueness of h in **HSig**. This shows that $\mathcal{L}$ together with the injections i' and i'' is the coproduct in **HLog** of $\mathcal{L}'$ and $\mathcal{L}''$. ◁

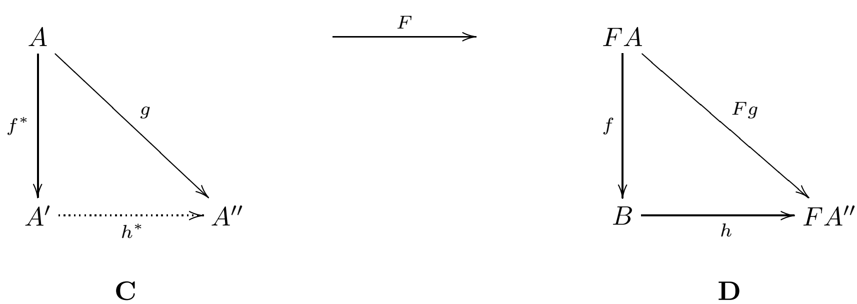


Figure 7.18: Cocartesian lifting of f through F

As an immediate consequence of Proposition 7.6.8 we have that the category **HLog** has finite coproducts. In order to define the constrained fibring in **HLog** we need to consider the (obvious) *forgetful functor* $Sg : \textbf{HLog} \to \textbf{HSig}$ and then prove that Sg has cocartesian liftings. For convenience of the reader, we start by briefly describing this notion, portrayed in Figure 7.18.

Definition 7.6.9 Let $F : \mathbf{C} \to \mathbf{D}$ be a functor, and let $f : F(A) \to B$ be a **D**-morphism. A *cocartesian lifting* (or *opcartesian lifting*) of f through F is a **C**-morphism $f^* : A \to A'$ such that:

- $F(f^*) = f$ (therefore $F(A') = B$);

- f^* satisfies the following universal property: if $g : A \to A''$ is a **C**-morphism and $h : B \to F(A'')$ is a **D**-morphism such that $h \circ f = F(g)$, there exists a unique **C**-morphism $h^* : A' \to A''$ with $F(h^*) = h$ and $h^* \circ f^* = g$.

The functor F is said to be a *cofibration* (or an *opfibration*) if every morphism $f : F(A) \to B$ admits a cocartesian lifting.

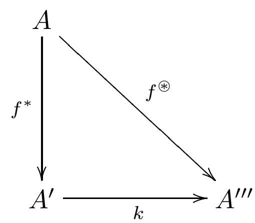


Figure 7.19: Unicity of the cocartesian lifting

Observe that, if $f^* : A \to A'$ and $f^{\circledast} : A \to A'''$ are cocartesian lifings for $f : F(A) \to B$ then there exists an isomorphism $k : A' \to A'''$ such that the diagram in Figure 7.19 commutes. This shows that the cocartesian lifting is unique up-to isomorphism.

We now return to constrained fibring. It is enough to prove the following result.

Proposition 7.6.10 *The forgetful functor Sg is a cofibration, that is, admits cocartesian liftings.*

Proof. Let $h : \Sigma \to \Sigma'$ be a signature morphism and let $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ be a logic system. Then, the logic system

$$h_{Sg}(\mathcal{L}) = \langle \Sigma', \mathcal{M}', h(R_g), h(R_\ell) \rangle$$

such that

$$\mathcal{M}' = \{M' \in Str(\Sigma') : M'|_h \in \mathcal{M}, \text{ and } M'|_h \in Ap(H) \text{ implies } M' \in Ap(h(H))\}$$

has the required universal property. That is,

$$\langle h_{Sg}(\mathcal{L}), h \rangle$$

is a cocartesian lifting of h by Sg at $\mathcal{L}$. In order to show this (and given that $Sg(f) = f$ for every morphism f in **HLog**), we must prove that h is indeed a morphism

$$h : \mathcal{L} \to h_{Sg}(\mathcal{L})$$

in **HLog**. Moreover, given a logic system $\check{\mathcal{L}} = \langle \check{\Sigma}, \check{\mathcal{M}}, \check{R}_g, \check{R}_\ell \rangle$, a logic morphism $g : \mathcal{L} \to \check{\mathcal{L}}$ and a signature morphism $f : \Sigma' \to \check{\Sigma}$ such that $f \circ h = g$, we must prove that f is indeed a morphism

$$f : h_{Sg}(\mathcal{L}) \to \check{\mathcal{L}}$$

in **HLog** such that $f \circ h = g$. More than this, f must be the unique morphism in **HLog** from $h_{Sg}(\mathcal{L})$ to $\check{\mathcal{L}}$ with such property. These facts can be easily proved in the same lines as the proof of Proposition 7.6.8, and the details are left to the reader. $\triangleleft$

Given a signature morphism $h : \Sigma \to \Sigma'$ and a logic system $\mathcal{L}$ defined over the signature Σ, we denote by $h_{Sg}(\mathcal{L})$ the codomain of the cocartesian lifting of h by Sg at $\mathcal{L}$ (as we did in the proof of Proposition 7.6.10). This construction is fundamental in order to define constrained fibrings, that is, fibring where we allow sharing of symbols. This construction is analogous to the constrained fibrings presented in the previous chapters, but adapted to the present category of logic systems.

Definition 7.6.11 Let Σ' and Σ'' two signatures. A *sharing constraint* over Σ' and Σ'' is a source diagram $\mathcal{G}$ in **HSig** of the form

$$\Sigma' \xleftarrow{h'} \check{\Sigma} \xrightarrow{h''} \Sigma''$$

for some signature $\check{\Sigma}$ and signature monomorphisms h' and h''. The pushout of the diagram $\mathcal{G}$ it will denoted by $\Sigma' \overset{\mathcal{G}}{\oplus} \Sigma''$. $\triangledown$

As said in Chapter 1, a pushout can be obtained as a coproduct followed by a coequalizer, whenever these constructions exist in the given category.

So, given a sharing constraint $\mathcal{G}$ in **HSig** as above, consider the diagram in Figure 7.20 where i' and i'' are the canonical injections associated to the coproduct $\Sigma' \oplus \Sigma''$, and

$$\Sigma' \overset{\mathcal{G}}{\oplus} \Sigma''$$

is the codomain of the coequalizer q of $i' \circ h'$ and $i'' \circ h''$. Then, the square in Figure 7.21 is the pushout of $\mathcal{G}$.

From these considerations, we define the constrained fibring in **HLog** as follows.

Definition 7.6.12 With notation as above, let $\mathcal{L}'$ and $\mathcal{L}''$ be two logic systems and let $\mathcal{G}$ be a sharing constraint over Σ' and Σ''. Then, their $\mathcal{G}$*-constrained fibring by sharing symbols* is the logic system

$$\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$$

given by $q_{Sg}(\mathcal{L}' \oplus \mathcal{L}'')$, where q is the coequalizer of $i' \circ h'$ and $i'' \circ h''$. $\triangledown$

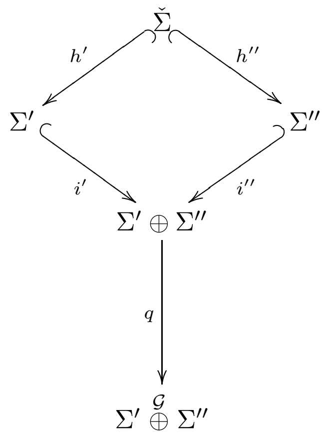


Figure 7.20: Obtaining a pushout of $\mathcal{G}$ using a coproduct and a coequalizer

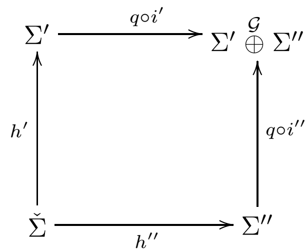


Figure 7.21: Pushout of $\mathcal{G}$

Since $Sg(\mathcal{L}' \oplus \mathcal{L}'') = Sg(\mathcal{L}') \oplus Sg(\mathcal{L}'') = \Sigma' \oplus \Sigma''$, then $\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$ is well-defined (see Figure 7.22).

Observe that, as usual, we can recover the unconstrained fibring as a special case of the constrained fibring by taking an appropriate sharing constraint: It is enough to take $\check{\Sigma}$ as the initial signature

$$\Sigma_0 = \langle R^0, F^0, Q^0 \rangle$$

such that

$$R^0_{\theta\theta'} = F^0_{\theta\theta'} = Q^0_{\theta\theta'\theta''} = \emptyset$$

for every $\theta, \theta', \theta'' \in \Theta(\mathbf{S})$; $h' : \Sigma_0 \to \Sigma'$ and $h'' : \Sigma_0 \to \Sigma''$ are the obvious (unique) morphisms.

When considering the forgetful functor

$$Sg : \mathbf{HLog} \to \mathbf{HSig},$$

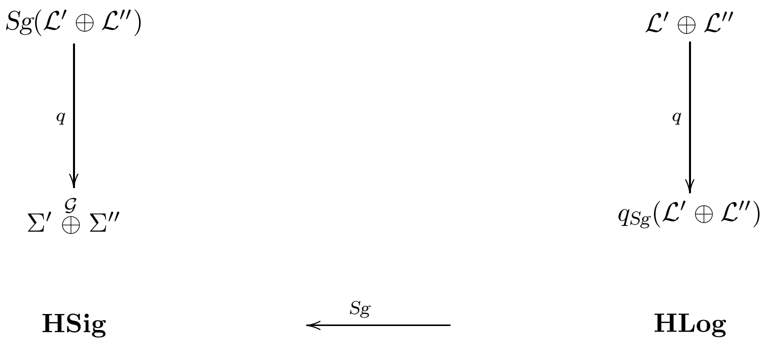


Figure 7.22: Cocartesian lifting of q

then $Sg(f) = f$ for every **HLog**-morphism f. The fibring

$$\mathcal{L} = q_{Sg}(\mathcal{L}' \oplus \mathcal{L}'')$$

of $\mathcal{L}'$ and $\mathcal{L}''$ is the "least" higher-order logic system defined by the signature $C = Sg(\mathcal{L})$ and the morphism q, in the following sense: given $\mathcal{L}_1$, a morphism $\mathcal{L}' \oplus \mathcal{L}'' \overset{q_1}{\to} \mathcal{L}_1$ in **HLog** and a morphism $C \overset{h}{\to} C_1$ in **HSig** such that $q_1 = h \circ q$, then h is, indeed, a **HLog**-morphism such that $q_1 = h \circ q$ in **HLog**, as Figure 7.23 shows.

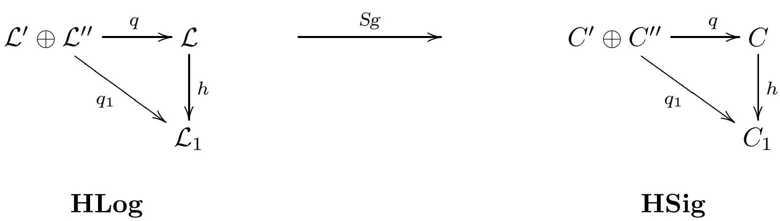


Figure 7.23: Universal property of constrained fibring

In particular, if $\mathcal{L}_1$ is defined over C such that q induces a morphism $\mathcal{L}' \oplus \mathcal{L}'' \overset{q}{\to} \mathcal{L}_1$ in **HLog** then the identity $C \overset{id_C}{\to} C$ induces a morphism $\mathcal{L} \overset{id_C}{\to} \mathcal{L}_1$ in **HLog**.

Remark 7.6.13 It is worth noting that the constrained fibring of two higher-order logic systems $\mathcal{L}'$ and $\mathcal{L}''$ is defined throughout the following steps: Starting from $\mathcal{L}'$ and $\mathcal{L}''$ in **HLog** and a source diagram $\mathcal{G}$ in **HSig** of the form

$$Sg(\mathcal{L}') \overset{h'}{\longleftarrow} \check{\Sigma} \overset{h''}{\longrightarrow} Sg(\mathcal{L}'')$$

(specifying the symbols to be shared in the process of fibring), where h' and h'' are monic, we compute the unconstrained fibring (coproduct) $\mathcal{L}' \oplus \mathcal{L}''$ in **HLog** (according to Definition 7.6.6) and then we obtain the morphism (coequalizer)

$$q : Sg(\mathcal{L}' \oplus \mathcal{L}'') \to Sg(\mathcal{L}') \overset{\mathcal{G}}{\oplus} Sg(\mathcal{L}'')$$

in **HSig** as described after Definition 7.6.11. Finally, we compute the (codomain of the) cocartesian lifting

$$q_{Sg}(\mathcal{L}' \oplus \mathcal{L}'')$$

of q by Sg at $\mathcal{L}' \oplus \mathcal{L}''$ as described in the proof of Proposition 7.6.10. This logic system, denoted by

$$\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$$

is the desired fibring of $\mathcal{L}'$ and $\mathcal{L}''$ by sharing $\mathcal{G}$. In Proposition 7.6.19 we will provide another characterization of the fibring. ∇

The universal construction of both forms of fibring as defined above is common to all the (syntactic and semantic) systems presented in the previous chapters: unconstrained fibring is always the coproduct of the given systems, and constrained fibring is always a cocartesian lifting with respect to the corresponding forgetful functor.

The next example serves as an illustration of the use of constrained fibring in **HLog** by sharing symbols. This example is also interesting because it is clarified the impact of choosing symbols as flexible or as rigid even in a logic (like *HOL*) where no modalities are available.

Example 7.6.14 We consider modal higher-order logic. Consider the fibring of $MPL_{\mathbf{K}}$ (introduced in Example 7.2.10) and *HOL* (introduced in Example 7.2.11) while sharing the propositional signature (as defined in Example 7.1.4) for obtaining a modal higher-order logic. Note that choosing a symbol as flexible or rigid in *HOL* changes nothing in that logic system. On the other hand, when *HOL* is combined with another logic system with modalities, rigid and flexible symbols will have quite different properties. For instance, in the resulting logic

$$((r \approx x) \Rightarrow (\Box\ (r \approx x)))$$

will be a theorem for any rigid symbol r, but not for a flexible symbol. ∇

Sometimes, besides sharing symbols, we may also want to share deduction rules. This form of combination appears as the *colimit* of a diagram in **HLog**. Moreover, the fibring by sharing symbols as introduced in Definition 7.6.12 can be characterized as an special case of fibring by sharing rules (see Proposition 7.6.19 below). In order to show this, we need to prove that **HLog** is small cocomplete. Using a well-known result from category theory, it is enough to show that **HLog** has small coproducts (the proof is similar to the proof of Proposition 7.6.8) and coequalizers. Observe that, given $h_1, h_2 : \Sigma \to \Sigma'$ in **HSig**, their coequalizer is $h : \Sigma' \to \Sigma''$ where $\Sigma'' = \langle \mathsf{R}'', \mathsf{F}'', \mathsf{Q}'' \rangle$ is the signature such that:

- $\mathsf{R}''_{\theta\theta'} = \mathsf{R}'_{\theta\theta'}/_{\cong^{\theta\theta'}_{\mathsf{R}'}}$;
- $\mathsf{F}''_{\theta\theta'} = \mathsf{F}'_{\theta\theta'}/_{\cong^{\theta\theta'}_{\mathsf{F}'}}$;
- $\mathsf{Q}''_{\theta\theta'\theta''} = \mathsf{Q}'_{\theta\theta'\theta''}/_{\cong^{\theta\theta'\theta''}_{\mathsf{Q}'}}$;

for every $\theta, \theta', \theta'' \in \Theta(\mathbf{S})$. Here, $\cong^{\theta\theta'}_{\mathsf{R}'} \subseteq (\mathsf{R}'_{\theta\theta'})^2$ is the least equivalence relation generated from

$$\{\langle h_1(r), h_2(r)\rangle : r \in \mathsf{R}_{\theta\theta'}\}.$$

The equivalence relations $\cong^{\theta\theta'}_{\mathsf{F}'}$ and $\cong^{\theta\theta'\theta''}_{\mathsf{Q}'}$ are defined in a similar way. The canonical morphism h is defined as $h(r') = [r']$ for each $r' \in \bigcup \mathsf{R}'$, $h(f') = [f']$ for each $f' \in \bigcup \mathsf{F}'$, and $h(q') = [q']$ for each $q' \in \bigcup \mathsf{Q}'$.

Proposition 7.6.15 *The category **HLog** has coequalizers.*

Proof. Let $h_1, h_2 : \mathcal{L} \to \mathcal{L}'$ be logic system morphisms. Let $h : Sg(\mathcal{L}') \to \Sigma''$ be the coequalizer in **HSig** of the morphisms $Sg(h_1), Sg(h_2) : Sg(\mathcal{L}) \to Sg(\mathcal{L}')$. For every $M' = \langle \mathcal{E}', W', \cdot_{M'}\rangle \in Str(\Sigma')$ let

$$[M'] = \langle \mathcal{E}', W', \cdot_{[M']}\rangle$$

such that $h(r')_{[M']} = r'_{M'}$ for every $r' \in \bigcup \mathsf{R}'$, $h(f')_{[M']} = f'_{M'}$ for every $f' \in \bigcup \mathsf{F}'$, and $h(q')_{[M']} = q'_{M'}$ for every $q' \in \bigcup \mathsf{Q}'$. Now, let $\mathcal{M}'_0$ be the class of all models $M' \in \mathcal{M}'$ such that:

- $r'_1 \cong^{\theta\theta'}_{\mathsf{R}'} r'_2$ implies $r'_{1M'} = r'_{2M'}$ for every $r'_1, r'_2 \in \mathsf{R}'_{\theta\theta'}$;
- $f'_1 \cong^{\theta\theta'}_{\mathsf{F}'} f'_2$ implies $f'_{1M'} = f'_{2M'}$ for every $f'_1, f'_2 \in \mathsf{F}'_{\theta\theta'}$;
- $q'_1 \cong^{\theta\theta'\theta''}_{\mathsf{Q}'} q'_2$ implies $q'_{1M'} = q'_{2M'}$ for every $q'_1, q'_2 \in \mathsf{Q}'_{\theta\theta'\theta''}$.

Note that, if $M' \in \mathcal{M}'_0$ then $[M'] \in Str(\Sigma'')$. Finally, let

$$\mathcal{M}'' = \{[M'] : M' \in \mathcal{M}'_0\}$$

and define $R''_g = h(R'_g)$ and $R''_\ell = h(R'_\ell)$. Then, the coequalizer in **HLog** of h_1, h_2 is given by $h : \mathcal{L}' \to \langle \Sigma'', \mathcal{M}'', R''_g, R''_\ell\rangle$. We leave to the reader the details of the proof. ◁

As mentioned above, from the existence in **HLog** of small coproducts and from Proposition 7.6.15 we obtain the desired result:

Corollary 7.6.16 *The category **HLog** is small cocomplete.*

In particular, the category **HLog** has pushouts. As already pointed out, pushouts are specially useful for combining two logics while sharing a common sublogic. In order to show an example of this claim, we need to prove a previous result (see Proposition 7.6.18 below).

Remark 7.6.17 Using the forgetful functor, it is possible to provide an alternative characterization of constrained fibring by sharing symbols, without referring to the cocartesian lifting. This is justified by the following fact:

Recall that, given a functor $F : \mathbf{C} \to \mathbf{D}$, the *left adjoint* of F (if it exists) is a functor $\bar{F} : \mathbf{D} \to \mathbf{C}$ such that there is a natural transformation

$$\eta : \mathrm{id}_{\mathbf{D}} \to F\bar{F}$$

satisfying the following property: for every $\mathbf{D}$-morphism $f : B \to FA$ there is a unique $\mathbf{C}$-morphism $g : \bar{F}B \to A$ such that $f = Fg \circ \eta_B$ in $\mathbf{D}$ (see Figure 7.24).

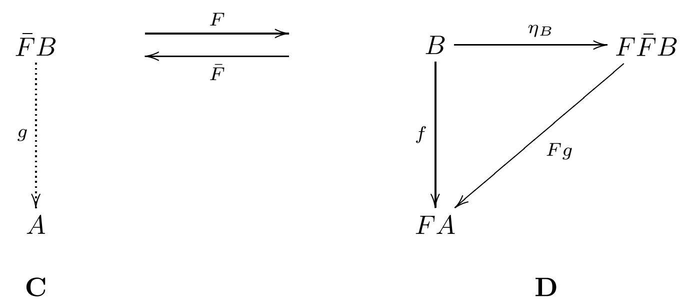


Figure 7.24: Adjointness property

Now, suppose that $F : \mathbf{C} \to \mathbf{D}$ is a functor which admits a left adjoint

$$\bar{F} : \mathbf{D} \to \mathbf{C}.$$

Assume also that both $\mathbf{C}$ and $\mathbf{D}$ have coproducts, which are preserved by F, that is:

$$F(A \oplus B) = F(A) \oplus F(B)$$

for every pair of objects A, B in $\mathbf{C}$. Suppose, additionally, that both $\mathbf{C}$ and $\mathbf{D}$ have coequalizers. Consider objects A and A' in $\mathbf{C}$, as well as $\mathbf{D}$-monomorphisms $i_1 : \bar{B} \to A$ and $i_2 : \bar{B} \to A'$, forming a source diagram (see Figure 7.25).

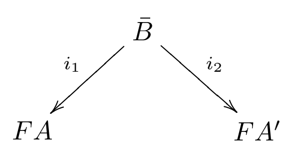


Figure 7.25: A source diagram in $\mathbf{D}$

Using the adjointness property and the hypothesis over $\mathbf{C}$, $\mathbf{D}$ and F, it can be obtained diagrams in $\mathbf{C}$ and $\mathbf{D}$ as depicted in Figure 7.26, where h_1, h_2 and

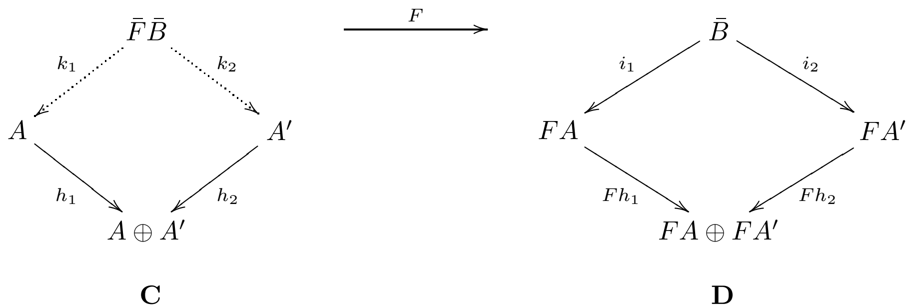


Figure 7.26: Coproducts obtained from a source diagram in **D**

Fh_1, Fh_2 are the canonical injections of the coproducts $A \oplus A'$ and $FA \oplus FA'$, respectively.

Let $q^* : A \oplus A' \to A''$ be the coequalizer of $h_1 \circ k_1$ and $h_2 \circ k_2$ in **C**, and let $q : FA \oplus FA' \to B$ be the coequalizer of $Fh_1 \circ i_1$ and $Fh_2 \circ i_2$ in **D** (see Figure 7.27).

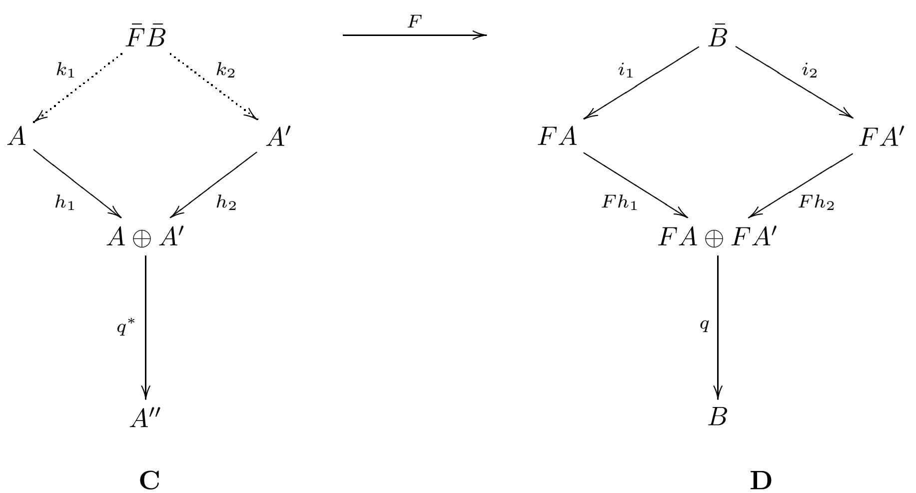


Figure 7.27: Coequalizers obtained from a source diagram in **D**

Then q^* is the cocartesian lifting of q, and so $B = FA''$. On the other hand, by construction q^* is the pushout in **C** of the source diagram in **C** displayed in Figure 7.28.

From this we see that the cocartesian lifting q^* obtained from the original source diagram in **D** can be replaced by an appropriate pushout in **C**. $\triangledown$

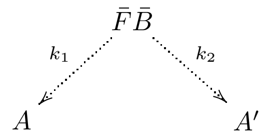


Figure 7.28: Derived source diagram in **C**

Proposition 7.6.18 *The forgetful functor Sg : **HLog** $\to$ **HSig** has a left adjoint G : **HSig** $\to$ **HLog**.*

Proof. Consider G : **HSig** $\to$ **HLog** such that $G(\Sigma) = \langle \Sigma, Str(\Sigma), \emptyset, \emptyset \rangle$ and $G(h) = h$. The reader can verify that G is indeed a left adjoint to Sg. ◁

Taking into account the above remark and the previous results about the functor Sg and the categories **HLog** and **HSig**, we obtain the following characterization of constrained fibring.

Proposition 7.6.19 *Given two logic systems $\mathcal{L}'$ and $\mathcal{L}''$ and a sharing constraint*

$$\mathcal{G} \;=\; Sg(\mathcal{L}') \xleftarrow{h'} \check{\Sigma} \xrightarrow{h''} Sg(\mathcal{L}'')$$

*in **HSig**, their $\mathcal{G}$-constrained fibring by sharing symbols is the pushout in **HLog** of*

$$\mathcal{L}' \xleftarrow{h'} G(\check{\Sigma}) \xrightarrow{h''} \mathcal{L}''.$$

Observe that Proposition 7.6.19 makes sense thanks to Corollary 7.6.16 and Proposition 7.6.18.

From the last result we can see that all forms of fibring appear as colimits in **HLog**. In view of this, from now on we shall establish results about colimits that will also apply to fibrings. To this end observe that, for every signature Σ, the logic system $G(\Sigma)$ is full and, thus, sound.

The results above can also be obtained for the other categories of logic systems studied in this book.

Notice that the category **HLog** might have been obtained as the flattening of the indexed category **HSig** $\to$ **Cat** (which can be easily defined). We refrained to analyze here the properties of this indexed category because we were interested only in the flat category of logic systems. However, it is worth noting that many properties of **HLog** would be derivable from interesting properties of the indexed category. The interested reader can consult [256] for relevant results about indexed categories.

We synthesize the properties of the fibring of higher-order based logics in the following way:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;

- *homogeneous combination mechanism at the semantic level*: both original logics are presented by interpretation structures;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fibring;
- *algorithmic combination of logics at the semantic level*: given the classes of interpretation structures for the original logics, we know how to define the class of interpretation structures for the fibring, but in many cases the given logics have to be pre-processed (that is, the interpretation structures for the original logics have to be extracted).

7.6.1 Preservation of soundness

Observe that all forms of combination of logic systems considered above automatically preserve soundness thanks to condition (2) in the definition of logic system morphism (recall Definition 7.6.1). In technical terms:

Theorem 7.6.20 *Sound logic systems are closed under colimits in* ***HLog***.

Proof. The first step is to prove that soundness is preserved by coproducts. For the sake of simplicity we just prove that the coproduct of two sound logic systems is sound. The general case is proven analogously and is left as an exercise to the reader.

Thus, let $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ be the coproduct of $\mathcal{L}'$ and $\mathcal{L}''$, and let $M \in \mathcal{M}$. By Proposition 7.6.8, $\mathcal{M}$ satisfies the properties listed in Definition 7.6.6 and then $M|_{i'} \in \mathcal{M}'$. Given that $\mathcal{M}' \subseteq Ap(H')$, because $\mathcal{L}'$ is sound by hypothesis, then $M|_{i'} \in Ap(H')$ and so

$$M \in Ap(i'(H'))$$

by definition of $\mathcal{M}$. Analogously we can prove that $M \in Ap(i''(H''))$ and so $M \in Ap(H)$, by definition of H. This shows that $\mathcal{M} \subseteq Ap(H)$ and then $\mathcal{L}$ is sound.

Finally, consider two parallel morphisms $h_1, h_2 : \mathcal{L} \to \mathcal{L}'$ such that $\mathcal{L}$ and $\mathcal{L}'$ are sound and let $h : \mathcal{L}' \to \mathcal{L}''$ be their coequalizer in **HLog** (recall the proof of Proposition 7.6.15). We have to show that $\mathcal{L}''$ is sound, that is, $\mathcal{M}'' \subseteq Ap(H'')$. Take $[M'] \in \mathcal{M}''$ and observe that $[M']|_h = M'$ and $M' \in Ap(H')$. Given that h is a morphism in **HLog** we obtain that

$$[M'] \in Ap(h(H')).$$

That is, $[M'] \in Ap(H'')$ (by definition of H'') and so $\mathcal{M}'' \subseteq Ap(H'')$. ◁

Corollary 7.6.21 *Both forms of fibring preserve soundness.*

7.6.2 Preservation of completeness

In this final subsection we address the problem of finding conditions to ensure the preservation of completeness by fibring.

It is not hard to find examples of complete logic systems such that their fibring is no longer complete.

Example 7.6.22 Completeness is not always preserved.
Consider the full logic systems $\mathcal{L}'$ and $\mathcal{L}''$ defined as follows.

- $\mathcal{L}' = \langle \Sigma', \mathcal{M}', R'_g, R'_\ell \rangle$ where $\Sigma' = \langle \mathsf{R}', \mathsf{F}', \mathsf{Q}' \rangle$ is such that all members of the families R', F' and Q' are empty, except $\mathsf{R}'_{\mathbf{i\Omega}} = \{p'\}$. On the other hand, $R'_\ell = R'_g = \{\langle \emptyset, p'(x), \mathbf{U} \rangle : x \in X_{\mathbf{i}}\}$.
- $\mathcal{L}'' = \langle \Sigma'', \mathcal{M}'', R''_g, R''_\ell \rangle$ where $\Sigma'' = \langle \mathsf{R}'', \mathsf{F}'', \mathsf{Q}'' \rangle$ is such that all members of the families R'', F'' and Q'' are empty, except $\mathsf{R}''_{\mathbf{1i}} = \{c''\}$ and $\mathsf{R}''_{\mathbf{1\Omega}} = \{\mathbf{t}''\}$. On the other hand, $R'_\ell = R'_g = \{\langle \emptyset, \mathbf{t}'', \mathbf{U} \rangle\}$.

Obviously the two logic systems are complete. However, their unconstrained fibring $\mathcal{L}$ is not complete. In fact, the resulting logic system is defined as follows: $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ such that

- $\Sigma = \langle \mathsf{R}, \mathsf{F}, \mathsf{Q} \rangle$, where all members of the families R, F and Q are empty, except $\mathsf{R}_{\mathbf{1i}} = \{c''\}$, $\mathsf{R}_{\mathbf{1\Omega}} = \{\mathbf{t}''\}$ and $\mathsf{R}_{\mathbf{i\Omega}} = \{p'\}$;
- $R_\ell = R_g = \{\langle \emptyset, p'(x), \mathbf{U} \rangle : x \in X_{\mathbf{i}}\} \cup \{\langle \emptyset, \mathbf{t}'', \mathbf{U} \rangle\}$.

The logic system $\mathcal{L}$ is still full: Preservation of fullness is a general property of fibring (see Corollary 7.6.25 below).

From this information it is easy to see that, in every interpretation structure $M \in \mathcal{M}$ over Σ, we have $[\![p'(c'')]\!]^M_{[]} = \mathit{true}_W$, therefore $\vDash^{\text{o}}_{\mathcal{L}} p'(c'')$. On the other hand, it is clear that $\nvdash^{\text{o}}_{\mathcal{L}} p'(c'')$. This shows that the (unconstrained) fibring $\mathcal{L}$ is not complete. ▽

The example above shows that, contrary to the case of soundness, completeness is not, in general, preserved by fibring.

However, following the idea of [282], it is possible to take advantage of a general completeness theorem in order to obtain a sufficient condition for the preservation of completeness by fibring. In the present case, the very general completeness theorem obtained at Section 7.5 is a canonical candidate. To start with, we need to state the following lemmas.

Lemma 7.6.23 *Let $h : \mathcal{L} \to \mathcal{L}'$ be a logic system morphism. Then for every $\Gamma \cup \{\delta\} \subseteq L(\Sigma, \vec{x})$, proviso Π and $o \in \{g, \ell\}$:*

$$\Gamma \vdash^{o\vec{x}}_{\mathcal{L}} \delta \lhd \Pi \text{ implies } h(\Gamma) \vdash^{o\vec{x}}_{\mathcal{L}'} h(\delta) \lhd \Pi.$$

Proof. We first prove the following technical result: given a proviso Π and a substitution σ over Σ, and defining the substitution σ' over Σ' by $\sigma' = \hat{h} \circ \sigma$, then $(\Pi\sigma') = (\Pi\sigma)$. To this end, let $\rho \in gSbs(\Sigma_1)$. Since $\hat{!}_{\Sigma'} \circ \hat{h} = \hat{!}_\Sigma$ then

$$\hat{!}_{\Sigma'} \circ \sigma' = \hat{!}_{\Sigma'} \circ (\hat{h} \circ \sigma) = (\hat{!}_{\Sigma'} \circ \hat{h}) \circ \sigma = \hat{!}_\Sigma \circ \sigma$$

and so

$$(\Pi\sigma')(\rho) = \Pi(\hat{\rho} \circ \hat{!}_{\Sigma'} \circ \sigma') = \Pi(\hat{\rho} \circ \hat{!}_\Sigma \circ \sigma) = (\Pi\sigma)(\rho).$$

This proves the intended result.

Now we can prove the main result. Suppose that $\Gamma \vdash^{g\vec{x}}_{\mathcal{L}} \delta \lhd \Pi$. Then we will show that $h(\Gamma) \vdash^{g\vec{x}}_{\mathcal{L}'} h(\delta) \lhd \Pi$ by induction on the length n of a global $\vec{x}$-derivation of δ from Γ with proviso Π.

Base $n = 1$. If $\delta \in \Gamma$ then the conclusion is obvious. On the other hand, if δ is obtained from an axiom $\langle \emptyset, \gamma, \Pi' \rangle$ in R_g using a substitution σ over Σ then δ is $\sigma(\gamma)$, $\Pi \le (\Pi'\sigma)$ and $\langle \emptyset, h(\gamma), \Pi' \rangle$ is an axiom in R'_g. Consider the substitution $\sigma' = \hat{h} \circ \sigma$ over Σ'. Then

$$\sigma'(h(\gamma)) = h(\sigma(\gamma))$$

by Lemma 7.6.7(1), and

$$(\Pi'\sigma') = (\Pi'\sigma)$$

by the result proved above. This shows the result for this case.

Step: Suppose that there are a rule

$$\langle \{\gamma'_1, \ldots, \gamma'_k\}, \delta', \Pi' \rangle$$

in R_g and a Σ substitution σ such that $\delta = \sigma(\delta')$ and $\delta_{i_j} = \sigma(\gamma'_j)$ for $j = 1, \ldots, k$ and some $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n-1\}$ and, moreover,

$$\Pi \le \Pi_{i_1} \sqcap \cdots \sqcap \Pi_{i_k} \sqcap (\Pi'\sigma).$$

By definition of global $\vec{x}$-derivation we immediately infer that $\Gamma \vdash^{g\vec{x}}_{\mathcal{L}} \delta_{i_j} \lhd \Pi_{i_j}$ for $j = 1, \ldots, k$. Therefore, by the induction hypothesis,

$$h(\Gamma) \vdash^{g\vec{x}}_{\mathcal{L}'} h(\delta_{i_j}) \lhd \Pi_{i_j} \quad \text{for } j = 1, \ldots, k.$$

But $\langle \{h(\gamma'_1), \ldots, h(\gamma'_k)\}, h(\delta'), \Pi' \rangle$ is in R'_g, by item (4) of the definition of logic system morphism. Hence, by considering the substitution $\sigma' = \hat{h} \circ \sigma$ over Σ' we obtain

$$\{\sigma'(h(\gamma'_1)), \ldots, \sigma'(h(\gamma'_k))\} \vdash^{g\vec{x}}_{\mathcal{L}'} \sigma'(h(\delta')) \lhd (\Pi'\sigma').$$

From this, using Lemma 7.6.7(1) (which says that $\sigma'(h(\gamma)) = h(\sigma(\gamma))$ for every $\gamma \in L(\Sigma)$) and by the result proved above we get

$$\{h(\delta_{i_1}), \ldots, h(\delta_{i_k})\} \vdash^{g\vec{x}}_{\mathcal{L}'} h(\delta) \lhd (\Pi'\sigma).$$

Therefore $h(\Gamma) \vdash^{g\vec{x}}_{\mathcal{L}'} h(\delta) \lhd \Pi$, by definition of global $\vec{x}$-derivation. The correspondent result for local derivations can be proved similarly. $\lhd$

Lemma 7.6.24 *Full logic systems are closed under colimits in* ***HLog***.

Proof. The first step is to show that fullness is preserved by coproducts. To simplify matters, we just prove that the coproduct of two full logic systems is full. The general case can be handled analogously.

Consider the coproduct $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$ of $\mathcal{L}'$ and $\mathcal{L}''$. Then, by Proposition 7.6.8 and Definition 7.6.6,

$$\begin{aligned}\mathcal{M} = \{M \in \mathit{Str}(\Sigma) : \; & M|_{i'} \in \mathcal{M}',\, M|_{i''} \in \mathcal{M}'',\\ & M|_{i'} \in \mathit{Ap}(H') \text{ implies } M \in \mathit{Ap}(i'(H')), \text{ and}\\ & M|_{i''} \in \mathit{Ap}(H'') \text{ implies } M \in \mathit{Ap}(i''(H''))\}.\end{aligned}$$

Since $\mathcal{M}' = \mathit{Ap}(H')$ and $\mathcal{M}'' = \mathit{Ap}(H'')$, by hypothesis, then

$$\begin{aligned}\mathcal{M} = \{M \in \mathit{Str}(\Sigma) : \; & M|_{i'} \in \mathit{Ap}(H'),\, M|_{i''} \in \mathit{Ap}(H''),\\ & M|_{i'} \in \mathit{Ap}(H') \text{ implies } M \in \mathit{Ap}(i'(H')), \text{ and}\\ & M|_{i''} \in \mathit{Ap}(H'') \text{ implies } M \in \mathit{Ap}(i''(H''))\}.\end{aligned}$$

Therefore

$$\begin{aligned}\mathcal{M} = \{M \in \mathit{Str}(\Sigma) : \; & M|_{i'} \in \mathit{Ap}(H'),\, M|_{i''} \in \mathit{Ap}(H''),\\ & M \in \mathit{Ap}(i'(H')), \text{ and } M \in \mathit{Ap}(i''(H''))\},\end{aligned}$$

that is, $\mathcal{M} = \mathit{Ap}(H)$, because of Lemma 7.6.7(3).

Finally, we will show that the codomain of the coequalizer is full whenever the logic systems in the given diagram are full.

Given parallel morphisms $h_1, h_2 : \mathcal{L} \to \mathcal{L}'$ such that $\mathcal{L}$ and $\mathcal{L}'$ are full, consider their coequalizer $h : \mathcal{L}' \to \mathcal{L}''$ in **HLog**, defined according to the proof of Proposition 7.6.15. We must show that

$$\mathcal{M}'' = \mathit{Ap}(H'').$$

By Proposition 7.6.20 we have that $\mathcal{L}''$ is sound and so $\mathcal{M}'' \subseteq \mathit{Ap}(H'')$. In order to show that $\mathit{Ap}(H'') \subseteq \mathcal{M}''$ consider $M'' \in \mathit{Ap}(H'')$ and let $M' = M''|_h$. By item (2) of Lemma 7.6.7 and the definition of H'' we get that $M' \in \mathit{Ap}(H')$ and so $M' \in \mathcal{M}'$, by fullness of $\mathcal{L}'$. On the other hand, by definition of reduct, and recalling the definition of $\mathcal{M}''$ given in the proof of Proposition 7.6.15, it is immediate to see that $M' \in \mathcal{M}'_0$. This shows that $M'' \in \mathcal{M}''$ as desired. $\triangleleft$

As a direct consequence of the last result we obtain the following:

Corollary 7.6.25 *Both forms of fibring preserve fullness.*

In view of this, we can put together the results obtained above and state a theorem of preservation of completeness by fibring, provided that the logic systems involved are strong enough.

Theorem 7.6.26 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be full logic systems with Hilbert calculi including HOL and with MTD. Then, for every sharing constraint $\mathcal{G}$ over Σ' and Σ'' such that all the symbols in Σ_{HOL} are shared, their fibring $\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$ is full, includes HOL, has MTD and is, therefore, complete.*

Proof. Let $\mathcal{L} = \mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$. Since both $\mathcal{L}'$ and $\mathcal{L}''$ are full and include *HOL* then $\mathcal{L}$ is full, by Lemma 7.6.24, and obviously it includes *HOL*. In order to prove that $\mathcal{L}$ is complete it is enough to show that it has *MTD*, by Proposition 7.5.4. And, by Proposition 7.5.2, it suffices to prove that

$$\{(\xi \Rightarrow \gamma_1), \ldots, (\xi \Rightarrow \gamma_k)\} \vdash^{\ell}_{\mathcal{L}} (\xi \Rightarrow \delta) \lhd \Pi$$

for every $\langle\{\gamma_1, \ldots, \gamma_k\}, \delta, \Pi\rangle \in R_\ell$, provided that the schema variable $\xi \in \Xi_{\mathbf{\Omega}}$ does not occur in the rule.

By definition of (constrained) fibring, every rule in $\mathcal{L}$ comes from $\mathcal{L}'$ or from $\mathcal{L}''$. Now assume that $\mathcal{L}'$ and $\mathcal{L}''$ have *MTD*. Given a local rule of $\mathcal{L}$, suppose that it comes from $\mathcal{L}'$ and it is of the form:

$$\langle\{i'(\gamma'_1), \ldots, i'(\gamma'_k)\}, i'(\delta'), \Pi'\rangle.$$

Since H' has *MTD* then

$$\{(\xi \Rightarrow \gamma'_1), \ldots, (\xi \Rightarrow \gamma'_k)\} \vdash^{\ell}_{\mathcal{L}'} (\xi \Rightarrow \delta') \lhd \Pi'$$

where ξ does not occur in the rule $\langle\{\gamma'_1, \ldots, \gamma'_k\}, \delta', \Pi'\rangle$ of $\mathcal{L}'$, by Proposition 7.5.2. By Lemma 7.6.23 we get

$$\{(\xi \Rightarrow i'(\gamma'_1)), \ldots, (\xi \Rightarrow i'(\gamma'_k))\} \vdash^{\ell}_{\mathcal{L}} (\xi \Rightarrow i'(\delta')) \lhd \Pi'$$

where $\xi \in \Xi_{\mathbf{\Omega}}$ does not occur in the given local rule of $\mathcal{L}$. A similar argument can be done if the given local rule of $\mathcal{L}$ comes from $\mathcal{L}''$. Then the result follows. $\lhd$

Proposition 7.6.26 is useful and interesting by its own. However, it requires that each of the given Hilbert calculi includes *HOL*. In order to cope with weaker logic systems we need to strength the last proposition. Thus, given two complete logic systems $\mathcal{L}_1$ and $\mathcal{L}_2$ (each of which does not include *HOL*) we could consider their respective enrichments with *HOL* (to be formally defined below) $\mathcal{L}_1^*$ and $\mathcal{L}_2^*$, and then try to compare the fibring of the original systems with the fibring of the enriched systems, observing that the latter is complete thanks to the proposition above.

To this end, we begin by introducing an appropriate notion of enrichment which will permit to add *HOL* to a given logic system. After this step, we establish a crucial property of the enrichment with *HOL*: The enrichment $\mathcal{L}^*$ of $\mathcal{L}$ is a conservative extension of $\mathcal{L}$ if and only if $\mathcal{L}$ is complete (under some weak conditions).

Definition 7.6.27 Given a logic system $\mathcal{L} = \langle \Sigma, \mathcal{M}, R_g, R_\ell \rangle$, consider the unconstrained fibring

$$\mathcal{L} \oplus HOL = \langle \check{\Sigma}, \check{\mathcal{M}}, \check{R}_g, \check{R}_\ell \rangle$$

of $\mathcal{L}$ and HOL with canonical injections $e : \Sigma \to \check{\Sigma}$ and $i : \Sigma_{HOL} \to \check{\Sigma}$. Denote by R_ℓ^{HOL} the set of local rules of HOL (recall Example 7.2.11). Then, the *HOL-enrichment* $\mathcal{L}^* = \langle \Sigma^*, \mathcal{M}^*, R_g^*, R_\ell^* \rangle$ of $\mathcal{L}$ is defined as follows:

- $\Sigma^* = \check{\Sigma}$;
- $\mathcal{M}^* = \check{\mathcal{M}}$;
- $R_g^* = \check{R}_g$;
- $R_\ell^* = \{\langle \emptyset, (\bigwedge \Gamma) \Rightarrow \delta, \Pi \rangle \;:\; \langle \Gamma, \delta, \Pi \rangle \in e(R_\ell)\} \cup i(R_\ell^{HOL})$. ∇

It is worth noting that $\mathcal{L}^*$ has *MTD*, because of Proposition 7.5.2. Moreover, $\Psi \vdash^{\mathrm{o}\vec{x}}_{\mathcal{L} \oplus HOL} \varphi$ implies $\Psi \vdash^{\mathrm{o}\vec{x}}_{\mathcal{L}^*} \varphi$, and the converse is true if and only if $\mathcal{L} \oplus HOL$ has *MTD*.

In the sequel it is convenient to use

$$e : \Sigma \to \Sigma^*$$

the embedding morphism of Σ into Σ^*. As expected, we say that $\mathcal{L}^*$ is a *conservative extension of* $\mathcal{L}$ if , for finite $\Psi \cup \{\varphi\} \subseteq gL(\Sigma)$ and $\mathrm{o} \in \{g, \ell\}$,

$$e(\Psi) \vdash^{\mathrm{o}}_{\mathcal{L}^*} e(\varphi) \text{ implies } \Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi.$$

Observe that, for $\mathrm{o} \in \{g, \ell\}$,

$$\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi \text{ implies } e(\Psi) \vdash^{\mathrm{o}}_{\mathcal{L}^*} e(\varphi)$$

because of the definition of $\mathcal{L}^*$.

It should be clear that, if every rule in $\mathcal{L}$ is robust (recall Definition 7.6.2), then every structure of $\mathcal{M}_{\mathcal{L}}$ appears in $\mathcal{M}^*$ with its interpretation map extended to the symbols of Σ_{HOL} as in $\mathcal{M}^0_{HOL}$ (recall Example 7.3.14). This means that no model of $\mathcal{L}$ is lost. Hence:

Lemma 7.6.28 *Let $\Psi \cup \{\varphi\} \subseteq gL(\Sigma, \vec{x})$ be a finite set and $o \in \{g, \ell\}$. Assume that $\mathcal{L}$ is robust. Then $\Psi \vDash^{\mathrm{o}\vec{x}}_{\mathcal{L}} \varphi$ if and only if $e(\Psi) \vDash^{\mathrm{o}\vec{x}}_{\mathcal{L}^*} e(\varphi)$.*

Proof. We start by observing that $[\![\psi]\!]_{\vec{x}}^{M^*|_e} = [\![\hat{e}(\psi)]\!]_{\vec{x}}^{M^*}$ for every $M^* \in Str(\Sigma^*)$ and $\psi \in gL(\Sigma)$. Consider $\mathcal{M}^*|_e = \{M^*|_e : M^* \in \mathcal{M}^*\}$. Therefore,

$$e(\Psi) \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma^*, \mathcal{M}^* \rangle} e(\varphi) \;\text{ if and only if }\; \Psi \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma, \mathcal{M}^*|_e \rangle} \varphi. \qquad (\dagger)$$

Since $e : \mathcal{L} \to \mathcal{L} \oplus HOL$ is a logic system morphism and $\mathcal{M}^* = \check{\mathcal{M}}$ then $\mathcal{M}^*|_e \subseteq \mathcal{M}$, by item (1) of Definition 7.6.1, and so: $\Psi \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma, \mathcal{M} \rangle} \varphi$ implies $\Psi \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma, \mathcal{M}^*|_e \rangle} \varphi$. But the latter implies that $e(\Psi) \vDash^{\mathrm{o}\vec{x}}_{\langle \Sigma^*, \mathcal{M}^* \rangle} e(\varphi)$, by (†).

Conversely, since $\mathcal{L}$ and *HOL* are assumed to be robust, we can easily prove that:

$$\mathcal{M}^* = \{M^* \in Str(\Sigma^*) \ : M^*|_e \in \mathcal{M} \text{ and } M^*|_i \in Ap(H_{HOL})\}.$$

On the other hand, note that every interpretation structure $M \in \mathcal{M}$ over Σ can be extended to a interpretation structure M^* over Σ^* such that $M^*|_e = M$ and $M^*|_i \in \mathcal{M}^0_{HOL}$. From this we get $\mathcal{M} = \mathcal{M}^*|_e$ and the result follows from (†). ◁

Lemma 7.6.29 *Let $\mathcal{L}$ be a full logic system. Then $\mathcal{L}^*$ is full and complete.*

Proof. As observed above, $\mathcal{L}^*$ has *MTD*. Since $\mathcal{L} \oplus HOL$ is full by Lemma 7.6.24, and $Ap_\ell(R^*_\ell) = Ap_\ell(\check{R}_\ell)$ (therefore $Ap(H_{\mathcal{L}^*}) = Ap(H_{\mathcal{L}\oplus HOL})$) we infer that $\mathcal{L}^*$ is also full. The result follows from Proposition 7.5.4. ◁

From the last two lemmas, and after introducing the next definition, we are finally ready to establish the result announced before about preservation of completeness by the result of fibring complete logic systems which do not contain *HOL*.

Definition 7.6.30 A logic system $\mathcal{L}$ is said to be *expressive* if for every context $\vec{x} = x_1 \ldots x_n$ there is a finite set $\Delta_{\vec{x}} \subseteq gL(\Sigma)$ such that the set of variables occurring free in $\Delta_{\vec{x}}$ is $\{x_1, \ldots, x_n\}$, and $\vdash^{\ell\vec{x}}_{\mathcal{L}} \varphi$ for every $\varphi \in \Delta_{\vec{x}}$. ▽

This technical condition will be used in the proof of item (2) of Proposition 7.6.31 below in order to obtain canonical contexts in local derivations. A sufficient condition for a logic system be expressive is to have a symbol $\approx_\theta$ for the (rigid) equality relation for every type θ, as well as an axiom $(x \approx_\theta x)$ establishing the reflexivity of the equality relation for every type θ.

Proposition 7.6.31 *Let $\mathcal{L}$ be a full logic system.*

1. *If $\mathcal{L}^*$ is a conservative extension of $\mathcal{L}$ then the logic system $\mathcal{L}$ is complete.*
2. *Assume that $\mathcal{L}$ is expressive and robust. If $\mathcal{L}$ is complete then $\mathcal{L}^*$ is a conservative extension of $\mathcal{L}$.*

Proof. 1. Suppose that $\mathcal{L}^*$ is a conservative extension of $\mathcal{L}$ and $\Psi \vDash^{\mathrm{o}}_{\mathcal{L}} \varphi$. Then,

$$e(\Psi) \vDash^{\mathrm{o}}_{\mathcal{L}^*} e(\varphi)$$

by the proof of the first part of Lemma 7.6.28 (which does not use the robustness of $\mathcal{L}$). Since $\mathcal{L}^*$ is complete, by Lemma 7.6.29, we obtain that $e(\Psi) \vdash^{\mathrm{o}}_{\mathcal{L}^*} e(\varphi)$ and so $\Psi \vdash^{\mathrm{o}}_{\mathcal{L}} \varphi$ because $\mathcal{L}^*$ conservatively extends $\mathcal{L}$. Therefore $\mathcal{L}$ is complete.

2. As a consequence of Lemma 7.6.29, since $\mathcal{L}$ is full then so is $\mathcal{L}^*$; in particular, $\mathcal{L}^*$ is sound. Consider a finite set $\Psi \cup \{\varphi\} \subseteq gL(\Sigma)$ such that $e(\Psi) \vdash^{\mathrm{o}}_{\mathcal{L}^*} e(\varphi)$. Then

$e(\Psi) \vdash^{\text{o}\vec{x}}_{\mathcal{L}^*} e(\varphi)$ for some context $\vec{x}$ and so $e(\Psi) \vDash^{\text{o}\vec{x}}_{\mathcal{L}^*} e(\varphi)$, because $\mathcal{L}^*$ is sound. By definition of semantic entailment we obtain

$$e(\Psi \cup \Delta) \vDash^{\text{o}\vec{x}}_{\mathcal{L}^*} e(\varphi)$$

for every finite set $\Delta \subseteq gL(\Sigma, \vec{x})$. In particular

$$e(\Psi \cup \Delta_{\vec{x}}) \vDash^{\text{o}\vec{x}}_{\mathcal{L}^*} e(\varphi)$$

where $\Delta_{\vec{x}}$ is a finite set of theorems of $\mathcal{L}$ associated with $\vec{x}$. This set exists because $\mathcal{L}$ is expressive. Thus, since $\mathcal{L}$ is robust, $\Psi \cup \Delta_{\vec{x}} \vDash^{\text{o}\vec{x}}_{\mathcal{L}} \varphi$, by Lemma 7.6.28. By construction of $\Delta_{\vec{x}}$ we get $\Psi \cup \Delta_{\vec{x}} \vDash^{\text{o}}_{\mathcal{L}} \varphi$. But $\mathcal{L}$ is complete, then $\Psi \cup \Delta_{\vec{x}} \vdash^{\text{o}}_{\mathcal{L}} \varphi$. Since every formula in $\Delta_{\vec{x}}$ is a theorem of $\mathcal{L}$ we obtain $\Psi \vdash^{\text{o}}_{\mathcal{L}} \varphi$. Therefore $\mathcal{L}^*$ is a conservative extension of $\mathcal{L}$. ◁

Using the last result we obtain an improvement of Proposition 7.6.26, which ensures the preservation of completeness by fibring without requiring the inclusion of *HOL*. However, this new result lies on the hypothesis of preservation of conservativeness of *HOL*-enrichment. More specifically:

Theorem 7.6.32 *Let $\mathcal{L}'$ and $\mathcal{L}''$ be full, expressive, robust and complete logic systems. Assume also that the conservativeness of*

$$(\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}'')^*$$

follows from the conservativeness of ${\mathcal{L}'}^*$ and ${\mathcal{L}''}^*$. Then, their fibring

$$\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}''$$

is also full, expressive, robust and complete.

It is an open problem to find conditions under which the conservativeness of

$$(\mathcal{L}' \overset{\mathcal{G}}{\oplus} \mathcal{L}'')^*$$

can be inferred from the conservativeness of ${\mathcal{L}'}^*$ and ${\mathcal{L}''}^*$. Some plausible requirements on each of the given logic systems $\mathcal{L}'$ and $\mathcal{L}''$ seem to be connected to the question of conservativeness of extensions by constants. For example, let ${\mathcal{L}'}_c$ be the system obtained from $\mathcal{L}'$ by adding a new constant to its signature. Then ${\mathcal{L}'}_c$ should be a conservative extension of $\mathcal{L}'$. Moreover, if ${\mathcal{L}'}^*$ is a conservative extension of $\mathcal{L}'$, then $({\mathcal{L}'}_c)^*$ should also be a conservative extension of ${\mathcal{L}'}_c$. The same of course should be required for $\mathcal{L}''$.

An evidence in favor of such conditions is the fact that the logic system $\mathcal{L}'$ of Example 7.6.22 does not satisfy the latter requirement. Indeed, if we add to Σ' the rigid constant $c'' \in \mathsf{R}''_{\mathbf{1i}}$ of Σ'' obtaining a logic system ${\mathcal{L}'}_{c''}$, then it is easy to see that $\vdash^{g}_{{\mathcal{L}'}^*_{c''}} p'(c'')$, despite $p'(c'')$ not being a theorem of ${\mathcal{L}'}_{c''}$.

7.7 Final remarks

In this chapter we have extended the fibring techniques to the context of higher-order logics with modalities. Following the trend in higher-order contexts, we adopted a categorial formulation namely using topos theory for setting up the semantic structures. We depart from this same context because we decided to use Hilbert calculus for deduction instead of the more common sequent calculus. This option was motivated by two reasons. The first one was homogeneity with the rest of the chapters. The second one was to avoid dealing with provisos like fresh variables and the like. We proved that for higher-order logics alone the adopted Hilbert calculus is complete.

Again, for proving preservation of completeness, we start by proving a general completeness theory for a given logic. This theorem involves a collection of sufficient conditions for completeness. The preservation consists in showing that those sufficient conditions are preserved by fibring. Hence, the fibring is complete using the general completeness theorem.

An immediate objective is to develop the same techniques using sequent calculus for deduction. We expect more work in what concerns provisos and maybe even in setting up the general completeness theorem.

At a certain stage we had to enrich *HOL*. We were not able to find sufficient conditions for the preservation of *HOL*-enrichment.

Chapter 8

Modulated fibring

Some combinations of logics, even when no symbols are shared, simply collapse to one of them. When one of them extends the other, often their combination just restores the differences between them. For instance, the unconstrained fibring of classical logic and intuitionistic logic collapses into classical logic, as it has been already pointed out in [82, 106]. This phenomenon, known as the collapsing problem, indicates that the notion of fibring considered so far imposes unwanted interconnections between the given logics. In fact, as we have seen before, when fibring interpretation systems an interpretation structure in the fibring should have the same truth-values and the same order relation as an interpretation structure in each of the components.

In this chapter we extend the notion of fibring presented in the previous chapters to the more powerful notion of modulated fibring. Modulated fibring allows a finer control of the combination, solving the collapsing problem both at the semantic and deductive levels. At the semantic level, the requirement that imposes the sharing of the truth value algebras of the two given logics is relaxed by giving as input to the fibring a translation between these algebras. This translation modulates the result. At the deductive level, this notion of modulated fibring leads to the existence of some provisos when applying the inference rules. Signatures are endowed with safe-relevant morphisms in order to restrict assignments and substitutions. Modulated fibring still preserves properties like soundness and completeness.

The collapsing problem is better understood with a semantic example in the setting of Chapter 3. Consider the semantic fibring of classical and intuitionistic logics with the following assumptions: (1) no symbols are shared; (2) classical logic is endowed with Boolean algebras and (3) intuitionistic logic is endowed with Heyting algebras. Consider the corresponding interpretation structures I' and I'' for classical and intuitionistic logic, respectively. The fibring $I' \cup I''$ contains all the ordered algebras such that the reduct to the propositional signature is an algebra in I' and the reduct to the intuitionistic signature is an algebra in I''. As a consequence both the intuitionistic implication and the intuitionistic negation become classic (observe that all Boolean algebras are Heyting algebras but not the other way around). Therefore we can say that the fibring is the disjoint union of two copies of classical logic even with no sharing of symbols.

This example is well known in the literature. By looking at it from a semantic point of view the problem is clear. The notion of fibring should be relaxed so that the algebras in the fibring should have carriers that although related to the carriers of the components cannot be the same.

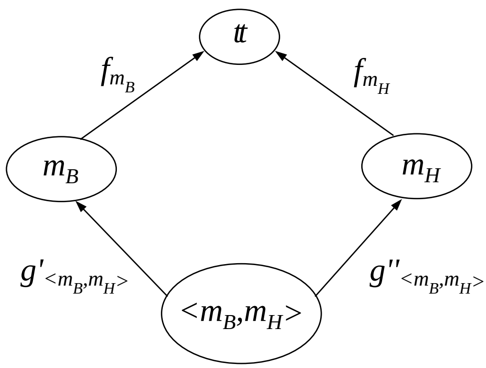


Figure 8.1: Relationship between models

In Figure 8.1, we represent a model of the fibring of classical and intuitionistic logics and the relationship with the original models. Let m_B be a model induced by a Boolean algebra and m_H be a model induced by a Heyting algebra. Assume that *tt* is a model whose carrier is a singleton. Then the corresponding model of the modulated fibring is $m = \langle m_B, m_H \rangle$. Moreover we must be able to relate it with the models of the components. The role of the model *tt* is to impose that the two models share a top value.

In Figure 8.2, we sketch the set of truth values B_m of m and its relationship with the sets of truth values B_{m_B} and B_{m_H} of m_B and m_H, respectively. The basic idea is that B_m is the disjoint union of B_{m_B} and B_{m_H}. Moreover, we are able to map each value of B_m to a point in B_{m_B} and a point in B_{m_H}. Take $b \in B_m$. Then if b is from B_{m_B} then it should be mapped to the corresponding point in B_{m_B} and to $\mathbf{t}_H$ (the top value of B_{m_H}). Similarly with respect to the case where b is from B_{m_H}. Moreover, the elements of both B_{m_B} and B_{m_H} are injected into B_m.

The chapter is structured as follows. In Section 8.1, we introduce the notions of modulated signature and modulated signature morphisms. In Section 8.2, we start by introducing modulated interpretation structures. Then we define modulated interpretation systems and the corresponding morphisms. Next we present the notion of bridge between modulated interpretation systems. Finally we describe the modulated fibring mechanism for these interpretation systems. We illustrate

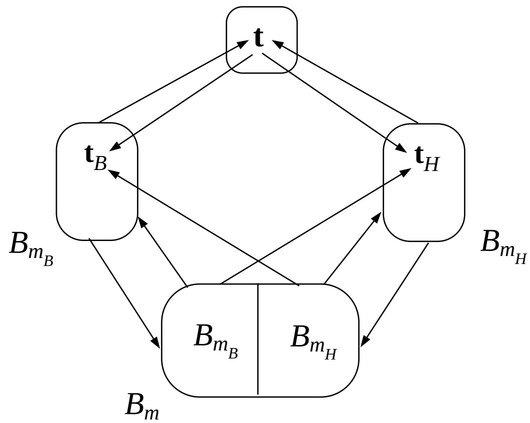


Figure 8.2: Relationship between truth-values

the concepts with examples including classical logic, intuitionistic logic, 3-valued Gödel and Łukasiewicz logics. In Section 8.3, we define modulated Hilbert calculus and their morphisms. Then we present the notion of bridge between two modulated Hilbert calculi and modulated fibring of these Hilbert calculi. In Section 8.4, we introduce the notion of modulated logic system and corresponding morphisms and then the usual notions of soundness and completeness. In Section 8.5, we discuss soundness and completeness preservation results. Finally, in Section 8.6 we give some final comments.

The present chapter capitalizes on the work developed in [243].

8.1 Language

Herein, we first introduce the notions of modulated pre-signature, modulated signature and the corresponding morphisms. The language over a modulated (pre-) signature then follows as usual.

Definition 8.1.1 A *modulated pre-signature* is a triple

$$\Sigma = \langle C, \&, \Xi \rangle$$

where C is a signature in the sense of Definition 2.1.1, & is a symbol and Ξ is a denumerable set of schema variables. ▽

The role of the symbol & will become clear later on when presenting the semantic structures. Moreover, this symbol is also essential for technical reasons in Subsection 8.5.2.

Definition 8.1.2 A *modulated pre-signature morphism*

$$h : \langle C, \&, \Xi\rangle \to \langle C', \&', \Xi'\rangle$$

is a pair $\langle h_1, h_2\rangle$ such that

- $h_1 = \{h_{1_k}\}_{k\in\mathbb{N}}$ is a family of maps $h_{1_k} : C_k \to C'_k$ for every $k \in \mathbb{N}$;
- $h_2 : \Xi \to \Xi'$ is a map. $\triangledown$

Prop/Definition 8.1.3 *Modulated pre-signatures and their morphisms constitute a category. This category has finite colimits.* $\triangledown$

Since the category of modulated pre-signatures and their morphisms has finite colimits thus, in particular, it has pushouts. A modulated pre-signature morphism $\langle h_1, h_2\rangle$ is said to be injective (surjective) whenever h_1 and every map of the family h_2 are injective (surjective).

Definition 8.1.4 A *modulated signature* Σ^+ is a co-cone in the category of modulated pre-signatures and their morphisms, that is,

$$\Sigma^+ = \langle C, \&, \Xi, S\rangle$$

where S is a set of modulated pre-signature morphisms whose destination is the pre-signature $\Sigma = \langle C, \&, \Xi\rangle$. $\triangledown$

In a modulated signature $\Sigma^+ = \langle C, \&, \Xi, S\rangle$, the morphisms $\breve{s} : \breve{\Sigma} \to \Sigma$ in the set S are called *safe-relevant* morphisms. Figure 8.3 provides an illustration of these morphisms.

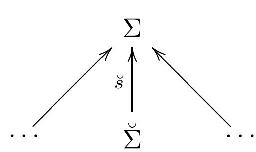


Figure 8.3: Safe-relevant morphisms diagram

From now on, for simplicity, we identify Σ^+ with Σ.

Safety will play an important role in the definition of the entailments by constraining the admissible assignments to schema variables in the range of safe-relevant morphisms. This is also the reason why the schema variables are, herein, local to signatures (that was not the case in the previous chapters, with the exception of Subsection 2.3.3 in Chapter 2). Safe-relevant morphisms will also be important in the definition of derivability in Hilbert calculi, where constraints are imposed to substitutions.

Definition 8.1.5 A *modulated signature morphism*

$$h : \langle C, \&, \Xi, S\rangle \to \langle C', \&', \Xi', S''\rangle$$

is a co-cone morphism, that is, h is a modulated pre-signature morphism such that $h \circ s \in S'$ whenever $s \in S$. ▽

Prop/Definition 8.1.6 *Modulated signatures and their morphisms constitute the category* ***mSig****. This category has finite colimits.* ▽

Since **mSig** has finite colimits, it has, in particular, pushouts.

Definition 8.1.7 The *language* over a modulated signature Σ is the set $L(\Sigma)$ of Σ-formulas defined as usual using the constructors in C, the schema variables in Ξ and the symbol & which is assumed to have arity 2. ▽

Each modulated signature morphism $h : \Sigma \to \Sigma'$ can be extended, as expected, to a map $h : L(\Sigma) \to L(\Sigma')$ between formulas noting that & translates to &′.

8.2 Modulated interpretation systems

We first present the modulated fibring mechanism from the semantic point of view. This notion of fibring was developed in an attempt to overcome collapsing problems that arise at the semantic level, and therefore the motivations and intuitions behind this mechanism were driven by the semantic aspects of the logics.

We start with the basic semantic unit: the modulated structure for a signature. Next we present the notions of modulated interpretation systems and their morphisms, denotation of formulas, and global and local entailments. Then, we introduce the notion of bridge between two modulated interpretation systems and we describe modulated fibring mechanism for modulated interpretation systems.

Recall that a pre-order over a set B is a binary relation on B that is reflexive and transitive. As usual we define the equivalence relation $\cong$ as follows:

$$b_1 \cong b_2 \text{ if and only if } b_1 \leq b_2 \text{ and } b_2 \leq b_1.$$

A pre-order over a set B is said to have finite meets whenever every finite subset B' of B has an infimum (a meet of B'). Note that finite meets exist if and only if the empty set and every set of cardinality 2 have meets. In a pre-order, meets are unique up to equivalence, and we use $b_1 \sqcap \ldots \sqcap b_k$ or $\sqcap\{b_1, \ldots, b_k\}$ for a choice of one of the meets of $\{b_1, \ldots, b_k\}$. Moreover, $\top$ denotes $\sqcap\emptyset$. Note that:

$$b \leq \top \text{ for every } b \in B.$$

Recall also that given two pre-orders over respectively B_1 and B_2, a map $h : B_1 \to B_2$ is said to preserve finite meets whenever $h(b_1 \sqcap \ldots \sqcap b_k)$ is a meet of $h(\{b_1, \ldots, b_k\})$, for every finite subset $\{b_1, \ldots, b_k\}$ of B_1. Note that $h : B_1 \to B_2$ preserves finite meets if and only if preserves meets of the empty set and meets of sets of cardinality 2.

In the sequel we consider modulated signatures $\Sigma = \langle C, \&, \Xi, S\rangle$, possibly with superscripts.

Definition 8.2.1 Consider the modulated signature Σ. A modulated *interpretation structure* over Σ is a triple

$$\mathcal{B} = \langle B, \leq, \nu\rangle$$

where $\langle B, \leq\rangle$ is a pre-order with finite meets and $\langle B, \nu\rangle$ is an algebra over C and & such that

- $\nu_2(\&)(b_1, b_2) = b_1 \sqcap b_2$;
- $\nu_k(c)(b_1, \ldots, b_k) \cong \nu_k(c)(d_1, \ldots, d_k)$ whenever $b_i \cong d_i$ for $i = 1, \ldots, k$. ∇

In the same spirit of Chapter 3, the elements in B are considered as truth values (or degrees). The first condition indicates that & behaves like a conjunction (whether or not such symbol is a constructor in the signature). The second condition is a congruence requirement. Clearly, modulated interpretation structures over Σ generalize the interpretation structures introduced in Definition 3.1.1 of Chapter 3.

In the sequel we omit the reference to the arity of the constructors and the subscripts in signature morphisms in order to make the notation lighter. Sometimes we also use $\vec{b}$ as a short hand for $b_1, \ldots, b_k$.

We now introduce modulated interpretation systems and their morphisms.

Definition 8.2.2 A *modulated interpretation system* is a tuple

$$I = \langle \Sigma, M, A\rangle$$

where Σ is a modulated signature, M is a class (of models), A is a map associating to each $m \in M$ a modulated interpretation structure $\mathcal{B}_m$ over Σ. ∇

Similar to what was done in Section 3.4 of Chapter 3, we include the class M because one can take the models of the logic at hand and use A to extract the underlying algebras. From this point of view, $A(M)$ is the class of modulated interpretation systems.

In the sequel, in the context of a modulated interpretation system $\langle \Sigma, M, A\rangle$, we always use $\mathcal{B}_m$ to denote the interpretation structure over Σ that A associates to a model m in M. This structure over Σ is assumed to be of the form

$$\langle B_m, \leq_m, \nu_m\rangle$$

and we will omit subscripts when no confusion arises. Similar remarks apply to $\top_m$ (meet of $\emptyset$) and $\cong_m$.

Some examples of modulated interpretation systems follow. They correspond to classic and intuitionistic propositional logics and some many-valued logics. In all examples the signature Σ is as follows: $\Sigma = \langle C, \&, \Xi, S\rangle$ where C_0 includes the symbol $\mathbf{t}$, $C_1 = \{\neg\}$, $C_2 = \{\wedge, \vee, \Rightarrow\}$, $C_k = \emptyset$ for all $k \geq 3$, $\&$ is $\wedge$, $\Xi = \{\xi_i : i \in \mathbb{N}\}$ and $S = \emptyset$. To better illustrate the modulated fibring mechanism, we consider herein classical conjunction and disjunction as primitive constructors.

Example 8.2.3 Recall Boolean algebras from Example 3.1.2. An interpretation system corresponding to classical logic is $\langle\Sigma, M, A\rangle$ where

- M is the class of all pairs $m = \langle \mathbb{B}, V\rangle$ where $\mathbb{B} = \langle B, \sqcap, \sqcup, -, \top, \bot\rangle$ is a Boolean algebra and $V : C_0 \to B$ is a map such that $V(\mathbf{t}) = \top$;
- $A(m) = \langle B, \leq, \nu\rangle$ where $\leq$ is defined as in Example 3.1.2 and ν extends the definition presented therein, that is, for every $b, b_1, b_2 \in B$,
  - $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
  - $\nu_0(c) = V(c)$ for every $c \in C_0$;
  - $\nu_1(\neg)(b) = -b$;
  - $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
  - $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
  - $\nu_2(\Rightarrow)(b_1 b_2) = (-b_1) \sqcup b_2$. ▽

Example 8.2.4 Recall Heyting algebras from Example 3.1.4. An interpretation system corresponding to intuitionistic logic is $\langle\Sigma, M, A\rangle$ where

- M is the class of all pairs $m = \langle \mathbb{B}, V\rangle$ where $\mathbb{B} = \langle B, \sqcap, \sqcup, \to, \top, \bot\rangle$ is a Heyting algebra and $V : C_0 \to B$ such that $V(\mathbf{t}) = \top$;
- $A(m) = \langle B, \leq, \nu\rangle$ where $\leq$ and ν are defined as in Example 3.1.4, that is, for every $b, b_1, b_2 \in B$
  - $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
  - $\nu_0(c) = V(c)$ for every $c \in C_0$;
  - $\nu_1(\neg)(b) = b \to \bot$;
  - $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
  - $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
  - $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \to b_2$. ▽

Example 8.2.5 Recall 3-valued Gödel algebras from Example 3.1.10. An interpretation system corresponding to 3-valued Gödel logic is $\langle\Sigma, M, A\rangle$ where

- M is the class of all pairs $m = \langle \mathbb{B}, V\rangle$ where $\mathbb{B} = \langle B, \sqcap, \sqcup, \sqsupset, \ominus, \top, \bot\rangle$ is a 3-valued Gödel algebra (recall that, up to isomorphisms, there is just one 3-valued Gödel algebra) and $V : C_0 \to B$ such that $V(\mathbf{t}) = \top$;
- $A(m) = \langle B, \leq, \nu\rangle$ where $\leq$ and ν are defined as in Example 3.1.10, that is, for every $b, b_1, b_2 \in B$
  - $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
  - $\nu_0(c) = V(c)$ for every $c \in C_0$;
  - $\nu_1(\neg)(b) = \ominus(b)$;
  - $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
  - $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
  - $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \sqsupset b_2$. ▽

Example 8.2.6 Recall 3-valued Łukasiewicz algebras from Example 3.1.9. An interpretation system corresponding to 3-valued Łukasiewicz logic is $\langle \Sigma, M, A\rangle$ where

- M is the class of all pairs $m = \langle \mathbb{B}, V\rangle$ where $\mathbb{B} = \langle B, \oplus, \ominus, \bot\rangle$ is a 3-valued Łukasiewicz algebra (recall that, up to isomorphisms, there is just one 3-valued Łukasiewicz algebra) and $V : C_0 \to B$ is a map;
- $A(m) = \langle B, \leq, \nu\rangle$ where $\leq$ is defined as in Example 3.1.9 and ν extends the definition presented therein, that is, for every $b, b_1, b_2 \in B$,
  - $b_1 \leq b_2$ if and only if $b_1 \sqcap b_2 = b_1$;
  - $\nu_0(c) = V(c)$ for every $c \in C_0$;
  - $\nu_1(\neg)(b) = \ominus b$;
  - $\nu_2(\wedge)(b_1, b_2) = b_1 \sqcap b_2$;
  - $\nu_2(\vee)(b_1, b_2) = b_1 \sqcup b_2$;
  - $\nu_2(\Rightarrow)(b_1, b_2) = b_1 \sqsupset b_2$. ▽

We now define modulated interpretation system morphisms. In the sequel, we consider modulated interpretation systems $I = \langle \Sigma, M, A\rangle$ with modulated signature $\Sigma = \langle C, \Xi, \&, S\rangle$, possibly with superscripts.

The next definition uses the notion of adjointness. We refer the reader to [66] for an account of adjointness in order structures.

Definition 8.2.7 A *modulated interpretation system morphism* $h : I \to I'$ is a tuple $\langle \hat{h}, \underline{h}, \dot{h}, \ddot{h}\rangle$ where:

- $\hat{h} : \Sigma \to \Sigma'$ is a morphism in **mSig**;

- $\underline{h} : M' \to M$ is a map;
- $\dot{h} = \{\dot{h}_{m'}\}_{m' \in M'}$ where $\dot{h}_{m'} : \langle B_{\underline{h}(m')}, \leq_{\underline{h}(m')} \rangle \to \langle B'_{m'}, \leq'_{m'} \rangle$ is a monotonic map;
- $\ddot{h} = \{\ddot{h}_{m'}\}_{m' \in M'}$ where $\ddot{h}_{m'} : \langle B'_{m'}, \leq'_{m'} \rangle \to \langle B_{\underline{h}(m')}, \leq_{\underline{h}(m')} \rangle$ is a monotonic map preserving finite meets;

such that

- $\ddot{h}_{m'}$ is left adjoint of $\dot{h}_{m'}$ for every $m' \in M'$, that is,

  $$b' \leq'_{m'} \dot{h}_{m'}(\ddot{h}_{m'}(b')) \text{ and } \ddot{h}_{m'}(\dot{h}_{m'}(b)) \leq_{\underline{h}(m')} b$$

  hold for every $b' \in B'_{m'}$ and $b \in B_{\underline{h}(m')}$;
- $\nu'_{m'}(\hat{h}(c))(\vec{b}') \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\vec{b}')))$ for every model $m' \in M'$, $c \in C_k$, $\vec{b}' \in (B'_{m'})^k$ and $k \in \mathbb{N}$. ∇

In the last item of the definition above, $\ddot{h}_{m'}(\vec{b}')$ stands for $(\ddot{h}_{m'}(b'_1), \ldots, \ddot{h}_{m'}(b'_k))$ for $\vec{b}' = (b'_1, \ldots, b'_k)$.

Note that the definition of left adjointness used above is equivalent to the usual one: $b' \leq'_{m'} \dot{h}_{m'}(b)$ if and only if $\ddot{h}_{m'}(b') \leq_{\underline{h}(m')} b$ for every $m' \in M'$, $b' \in B'_{m'}$ and $b \in B_{\underline{h}(m')}$.

A modulated interpretation system morphism $h : I \to I'$ has four components. The first component is a modulated signature morphism $\hat{h}$ relating the signatures involved. The second component is a (contravariant) map $\underline{h}$ relating the classes of models. For a diagrammatic perspective see Figure 8.4.

$$I \xrightarrow{\;h\;} I'$$

$$\Sigma \xrightarrow{\;\hat{h}\;} \Sigma' \qquad\qquad M \xleftarrow{\;\underline{h}\;} M'$$

Figure 8.4: Components of interpretation system morphism

The two last components are the families $\dot{h}$ and $\ddot{h}$ of maps, indexed by the class of models in I', that relate truth values of interpretation structures over Σ to truth values of interpretation structures over Σ' and vice-versa: for each model m', $\dot{h}_{m'}$ associates a truth value of the interpretation structure $\mathcal{B}_{\underline{h}(m')}$ over Σ to a truth value of the structure $\mathcal{B}'_{m'}$ over Σ', while $\ddot{h}_{m'}$ associates a truth value of $\mathcal{B}'_{m'}$ to a truth value of $\mathcal{B}_{\underline{h}(m')}$. Figure 8.5 illustrates these components.

Since $\ddot{h}_{m'}$ is left adjoint of $\dot{h}_{m'}$, we can conclude that $\dot{h}_{m'}$ also preserves finite meets for every $m' \in M'$ (see Lemma 8.2.9). The last condition indicates that

$$B_{\underline{h}(m')} \xrightarrow{\dot{h}_{m'}} B'_{m'} \qquad\qquad B_{\underline{h}(m')} \xleftarrow{\ddot{h}_{m'}} B'_{m'}$$

Figure 8.5: Relationship between truth-value sets

denotations of constructors from C in a model m' can be given for any truth values in $B'_{m'}$ by using the two maps.

Note that the notion of morphism between interpretation systems presented in [282], and described in Section 3.4 of Chapter 3, is the particular case of the one presented above with $\dot{h}_{m'} = \mathrm{id}_{B'_{m'}}$, $\ddot{h}_{m'} = \mathrm{id}_{B_{\underline{h}(m')}}$ and hence, $B_{\underline{h}(m')} = B'_{m'}$, etc.

Prop/Definition 8.2.8 *Modulated interpretation systems and their morphisms constitute the category* ***mInt****.*

The following facts about modulated interpretation system morphisms will be useful later on.

Lemma 8.2.9 *Let $h : I \to I'$ be a modulated interpretation system morphism.*

1. *$\dot{h}_{m'}$ preserves finite meets, for every $m' \in M'$.*
2. *$\nu'_{m'}(\&')(\dot{h}_{m'}(b_1), \dot{h}_{m'}(b_2)) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(\&)(b_1, b_2))$, for every $m' \in M'$ and $b_1, b_2 \in B_{\underline{h}(m')}$.*
3. *$\ddot{h}_{m'}(\dot{h}_{m'}(b)) \cong_{\underline{h}(m')} b$, for every $m' \in M'$ and $b \in B_{\underline{h}(m')}$, whenever $\ddot{h}_{m'}$ is surjective.*

Proof.

1. From the left adjoint condition, $\dot{h}$ has left adjoint. Hence $\dot{h}$ preserves all meets.

2. Use the definitions of $\nu_{\underline{h}(m')}(\&)$ and $\nu'_{m'}(\&')$ and 1.

3. The left adjoint condition ensures that $\ddot{h}_{m'}(\dot{h}_{m'}(b)) \leq_{\underline{h}(m')} b$. Since $\ddot{h}_{m'}$ is surjective, there is $b' \in B'_{m'}$ such that $b = \ddot{h}_{m'}(b')$. The left adjoint condition ensures that $b' \leq'_{m'} \dot{h}_{m'}(\ddot{h}_{m'}(b'))$, that is, $b' \leq'_{m'} \dot{h}_{m'}(b)$. By monotonicity,

$$\ddot{h}_{m'}(b') \leq_{\underline{h}(m'} \ddot{h}_{m'}(\dot{h}_{m'}(b))$$

that is, $b \leq_{\underline{h}(m'} \ddot{h}_{m'}(\dot{h}_{m'}(b))$. ◁

Before defining the denotation of formulas, we first have to introduce the notion of assignment. Herein, we have a special kind of assignments, the safe assignments. They give special values to schema variables in the codomain of safe-relevant signature morphisms. To define safe assignments we need to consider a particular (sub)algebra for each modulated signature morphism: if $s : \breve{\Sigma} \to \Sigma$

is a modulated signature morphism and $\mathcal{B}$ is an interpretation structure over Σ, $\mathcal{B}(s) = \langle B(s), \leq_{\mathcal{B}(s)}, \nu_{\mathcal{B}(s)} \rangle$ is the smallest subalgebra of $\mathcal{B}$ for the signature $s(\breve{\Sigma})$ such that $b_1 \in B(s)$ whenever $b_1 \cong b_2$ with $b_1 \in B$ and $b_2 \in B(s)$. In the sequel, whenever we refer to a modulated signature morphism $s : \breve{\Sigma} \to \Sigma$, we always assume that $\breve{\Sigma} = \langle \breve{C}, \breve{\&}, \breve{\Xi}, \breve{S} \rangle$. The following lemma will be useful later on.

Lemma 8.2.10 *Let $h : I \to I'$ be a modulated interpretation system morphism.*

1. *For each $b' \in B'_{m'}(\hat{h})$ there exists $b \in B_{\underline{h}(m')}$ such that $b' \cong_{m'} \dot{h}_{m'}(b)$.*

2. *If $\ddot{h}$ is surjective and $s : \breve{\Sigma} \to \Sigma$ is a modulated signature morphism then for each $b' \in B'_{m'}(\hat{h} \circ s)$ there exists $b \in B_{\underline{h}(m')}(s)$ such that $b' \cong_{m'} \dot{h}_{m'}(b)$.*

Proof. Note that if $f : (C^1, A^1, \Xi^1, S^1) \to (C^2, A^2, \Xi^2, S^2)$ is a modulated signature morphism and $b \in B^2_m(f)$ then $b \cong_m \nu^2_m(f(c_0))$ where $c_0 \in C^1_0$, or $b \cong_m \nu^2_m(f(c))(\vec{b})$ where $c \in C^2_k$, $k > 0$ and $\vec{b} \in (B^2_m(f))^k$, or $b \cong_m \nu^2_m(\&^2)(b_1, b_2)$ (that is, $b \cong_m b_1 \sqcap b_2$) where $b_1, b_2 \in B^2_m(f)$.

1. The result follows by induction.

Base: Let $b' \cong_{m'} \nu'_{m'}(\hat{h}(c_0))$. Using Definition 8.2.7,

$$\nu'_{m'}(\hat{h}(c_0)) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(c_0))$$

and the result follows by transitivity.

Step: The case $b' \cong_{m'} \nu'_{m'}(\hat{h}(c))(\vec{b'})$ where $c \in C_k$, $k > 0$ and $\vec{b'} \in (B'_{m'}(\hat{h}))^k$, is similar to the base. Let $b' \cong_{m'} b'_1 \sqcap b'_2$. By the induction hypothesis,

$$b'_1 \cong_{m'} \dot{h}_{m'}(b_1) \text{ and } b'_2 \cong_{m'} \dot{h}_{m'}(b_2)$$

for some $b_1, b_2 \in B_{\underline{h}(m')}$. Hence, $b'_1 \sqcap b'_2 \cong_{m'} \dot{h}_{m'}(b_1) \sqcap \dot{h}_{m'}(b_2)$. By 1 of Lemma 8.2.9, $\dot{h}_{m'}(b_1) \sqcap \dot{h}_{m'}(b_2) \cong_{m'} \dot{h}_{m'}(b_1 \sqcap b_2)$, and the result follows using transitivity.

2. The result follows by induction. Note that $\hat{h} \circ s$ is a modulated signature morphism.

Base: Let $b' \cong_{m'} \nu'_{m'}(\hat{h}(s(\breve{c}_0)))$. Using Definition 8.2.7,

$$\nu'_{m'}(\hat{h}(s(\breve{c}_0))) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(s(\breve{c}_0))).$$

The result follows by transitivity, since $\nu_{\underline{h}(m')}(s(\breve{c}_0)) \in B_{\underline{h}(m')}(s)$.

Step: Let $b' \cong_{m'} \nu'_{m'}(\hat{h}(s(\breve{c})))(\vec{b'})$ where $c \in C_k$, $k > 0$ and $\vec{b'} \in (B'_{m'}(\hat{h} \circ s))^k$. By the induction hypothesis, $b'_i \cong_{m'} \dot{h}_{m'}(b_i)$, where $b_i \in B_{\underline{h}(m')}(s)$, for each b'_i in $\vec{b'}$. Using Definition 8.2.1,

$$\nu'_{m'}(\hat{h}(s(\breve{c})))(\vec{b'}) \cong_{m'} \nu'_{m'}(\hat{h}(s(\breve{c})))(\dot{h}_{m'}(\vec{b}))$$

and, using Definition 8.2.7,

$$\nu'_{m'}(\hat{h}(s(\breve{c})))(\dot{h}_{m'}(\vec{b})) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(s(\breve{c}))(\ddot{h}_{m'}(\dot{h}_{m'}(\vec{b})))).$$

By 3 of Lemma 8.2.9, $\ddot{h}_{m'}(\dot{h}_{m'}(\vec{b})) \cong_{\underline{h}(m')} \vec{b}$. The result follows using again Definitions 8.2.1, 8.2.7 and transitivity. The proof of the case $b' \cong_{m'} \nu'_{m'}(\hat{h}(s(\breve{\&})))(b'_1, b'_2)$, that is, $b' \cong_{m'} b'_1 \sqcap b'_2$, is similar to the one presented in 1. $\triangleleft$

We recall the notion of assignment given in Chapter 3 and introduce the concept of safe assignment. In the sequel, given a formula $\varphi \in L(\Sigma)$, we use $\mathrm{Var}(\varphi)$ to denote the set of schema variables occurring in φ. Similarly with respect to a set of formulas $\Gamma \subseteq L(\Sigma)$.

Definition 8.2.11 Let $\Sigma = \langle C, \&, \Xi, S\rangle$ be a modulated signature. An *assignment* over a modulated interpretation structure $\mathcal{B}$ over Σ is a map $\alpha : \Xi \to B$. The assignment α is said to be *safe* for a set of formulas $\Gamma \subseteq L(\Sigma)$ whenever $\alpha(s(\breve{\xi})) \in B(s)$ for every $s : \breve{\Sigma} \to \Sigma$ in S and $\breve{\xi} \in \breve{\Xi}$ such that $s(\breve{\xi}) \in \mathrm{Var}(\Gamma)$. ∇

Whenever no confusion arises, when dealing with assignments we simply omit the reference to the modulated interpretation structure over Σ. Safe assignments $\alpha : \Xi \to B$ for a set $\Gamma \subseteq L(\Sigma)$ ensure that only special truth values are associated to particular schema variables occurring in formulas in Γ. These particular schema variables are images, through safe-relevant morphisms $s : \breve{\Sigma} \to \Sigma$, of schema variables in $\breve{\Sigma}$ and the truth values that α assigns to them are always truth values in the subset $B(s)$ of B. As we will see below, only safe assignments are relevant when defining global and local entailments.

Given a modulated interpretation structure $\mathcal{B} = \langle B, \leq, \nu\rangle$ over Σ, and an assignment α over $\mathcal{B}$, the denotation map $[\![\cdot]\!]^{\alpha}_{\mathcal{B}} : L(\Sigma) \to B$ is defined as usual using α and ν.

Definition 8.2.12 Let $\mathcal{B} = \langle B, \leq, \nu\rangle$ be a modulated interpretation structure over Σ.

A formula $\varphi \in L(\Sigma)$ is *globally satisfied* by $\mathcal{B}$ and a safe assignment α for φ, written $\mathcal{B}\alpha \Vdash \varphi$, whenever $[\![\varphi]\!]^{\alpha}_{\mathcal{B}} \cong \top$.

A formula $\varphi \in L(\Sigma)$ is *locally satisfied* by $\mathcal{B}$, a safe assignment α for φ and $b \in B$, written $\mathcal{B}\alpha b \Vdash \varphi$, whenever $b \leq [\![\varphi]\!]^{\alpha}_{\mathcal{B}}$. ∇

In the context of a modulated interpretation system $\langle \Sigma, M, A\rangle$ and given a model m in M, for simplicity, we refer to an assignment over m, instead of an assignment over $\mathcal{B}_m$, and we will use $[\![\cdot]\!]^{\alpha}_{m}$ instead of $[\![\cdot]\!]^{\alpha}_{\mathcal{B}_m}$. We also write $m\alpha \Vdash \varphi$ and $m\alpha b \Vdash \varphi$ whenever $\mathcal{B}_m\alpha \Vdash \varphi$ and $\mathcal{B}_m\alpha b \Vdash \varphi$, respectively.

Next, we introduce the notions of global entailment and local entailment in a modulated interpretation system.

Definition 8.2.13 Let I be a modulated interpretation system.

A formula $\varphi \in L(\Sigma)$ is globally entailed by a finite set $\Psi \subseteq L(\Sigma)$ in I if, for every model $m \in M$ and assignment α over $\mathcal{B}_m$ safe for $\Psi \cup \{\varphi\}$, $m\alpha \Vdash \varphi$ whenever

$m\alpha \Vdash \psi$ for every $\psi \in \Psi$. A formula $\varphi \in L(\Sigma)$ is *globally entailed* by $\Gamma \subseteq L(\Sigma)$ in I, written

$$\Gamma \vDash^g_I \varphi$$

if there is a finite set $\Psi \subseteq \Gamma$ that globally entails φ.

A formula $\varphi \in L(\Sigma)$ is locally entailed by a finite set $\Psi \subseteq L(\Sigma)$ in I if, for every model $m \in M$, $b \in B_m$ and assignment α over $\mathcal{B}_m$ safe for $\Psi \cup \{\varphi\}$, $m\alpha b \Vdash \varphi$ whenever $m\alpha b \Vdash \psi$ for every $\psi \in \Psi$. A formula $\varphi \in L(\Sigma)$ is *locally entailed* by $\Gamma \subseteq L(\Sigma)$ in I, written

$$\Gamma \vDash^\ell_I \varphi$$

if there is a finite set $\Psi \subseteq \Gamma$ that locally entails φ. $\triangledown$

We provide a necessary and sufficient condition for local entailment from a finite set of formulas. Note that local entailment has already been characterized in similar terms in the previous chapters.

Proposition 8.2.14 *Let I be a modulated interpretation system, Γ be a finite subset of $L(\Sigma)$ and φ in $L(\Sigma)$. Then*

$$\Gamma \vDash^\ell_I \varphi \text{ if and only if } \sqcap\{[\![\gamma]\!]^\alpha_m : \gamma \in \Gamma\} \leq [\![\varphi]\!]^\alpha_m$$

for every model $m \in M$ and safe assignment α over m for $\Gamma \cup \{\varphi\}$.

Proof. Assume that $\Gamma \vDash^\ell_I \varphi$. Let

$$b = \sqcap\{[\![\gamma]\!]^\alpha_m : \gamma \in \Gamma\}.$$

Then $b \leq [\![\gamma]\!]^\alpha_m$ for each $\gamma \in \Gamma$. Since $\Gamma \vDash^\ell_I \varphi$, it follows that $b \leq [\![\varphi]\!]^\alpha_m$. Conversely, assume that $\sqcap\{[\![\gamma]\!]^\alpha_m : \gamma \in \Gamma\} \leq [\![\varphi]\!]^\alpha_m$ for every $m \in M$ and α over m. Given $m \in M$, $b \in B_m$ and α over m, if $b \leq [\![\gamma]\!]^\alpha_m$ for each $\gamma \in \Gamma$, then $b \leq \sqcap\{[\![\gamma]\!]^\alpha_m : \gamma \in \Gamma\}$. Since $\sqcap\{[\![\gamma]\!]^\alpha_m : \gamma \in \Gamma\} \leq [\![\varphi]\!]^\alpha_m$, by transitivity, $b \leq [\![\varphi]\!]^\alpha_m$. Hence, $\Gamma \vDash^\ell_I \varphi$. $\triangleleft$

Global entailment and local entailment are both preserved by particular modulated interpretation system morphisms. Before presenting the corresponding proof, we need the following lemma that relates denotations of formulas with the denotations of their translations.

Lemma 8.2.15 *Let $h : I \to I'$ be a modulated interpretation system morphism such that $\ddot{h}_{m'}$ is surjective for every $m' \in M'$. Consider the assignment α' over m' and the assignment $\underline{h}(\alpha') : \Xi \to B_{\underline{h}(m')}$ such that $\underline{h}(\alpha')(\xi) = \ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))$.*

1. $[\![\hat{h}(\xi)]\!]^{\alpha'}_{m'} \cong_{m'} \dot{h}_{m'}([\![\xi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')})$ *if α' is safe for $\hat{h}(\xi)$, $\xi \in \Xi$, and $\hat{h} \in S'$.*

2. $[\![\hat{h}(\varphi)]\!]^{\alpha'}_{m'} \cong_{m'} \dot{h}_{m'}([\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')})$*, for every ground formula $\varphi \in L(\Sigma)$.*

3. $[\![\hat{h}(\varphi)]\!]^{\alpha'}_{m'} \cong_{m'} \dot{h}_{m'}([\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')})$ *if* $\varphi \in L(\Sigma)$, α' *is safe for* $\hat{h}(\varphi)$ *and* $\hat{h} \in S'$.

Proof.

1. Since α' is safe for $\hat{h}(\xi)$ and $\hat{h} \in S'$, $[\![\hat{h}(\xi)]\!]^{\alpha'}_{m'} = \alpha'(\hat{h}(\xi)) \in B'(\hat{h})$. Hence, we have that (i) $\alpha'(\hat{h}(\xi)) \cong_{m'} \nu'_{m'}(\hat{h}(c))(\vec{b'})$ for some $c \in C_k$ and $\vec{b'} \in (B'(\hat{h}))^k$ with $k \in \mathbb{N}$ or (ii) $\alpha'(\hat{h}(\xi)) \cong_{m'} \nu'_{m'}(\&')(b'_1, b'_2)$ for some $b'_1, b'_2 \in B'(\hat{h})$, that is, $\alpha'(\hat{h}(\xi)) \cong_{m'} b'_1 \sqcap b'_2$. Note that

$$\begin{aligned} \dot{h}_{m'}([\![\xi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}) &= \dot{h}_{m'}(\underline{h}(\alpha')(\xi)) \\ &= \dot{h}_{m'}(\ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))). \end{aligned}$$

In case (i), by Definition 8.2.7 and transitivity,

$$\alpha'(\hat{h}(\xi)) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\vec{b'}))).$$

Hence, using monotonicity,

$$\ddot{h}_{m'}(\alpha'(\hat{h}(\xi))) \cong_{m'} \ddot{h}_{m'}(\dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\vec{b'})))).$$

Using 3 of Lemma 8.2.9 and transitivity,

$$\ddot{h}_{m'}(\alpha'(\hat{h}(\xi))) \cong_{m'} \nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\vec{b'})).$$

Therefore, using monotonicity,

$$\dot{h}_{m'}(\ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))) \cong_{m'} \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\vec{b'}))).$$

By transitivity, $\dot{h}_{m'}(\ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))) \cong_{m'} \alpha'(\hat{h}(\xi))$ and we are done.
In case (ii), by Lemma 8.2.10, it follows that $b'_1 \cong_{m'} \dot{h}_{m'}(b_1)$ and $b'_2 \cong_{m'} \dot{h}_{m'}(b_2)$ for $b_1, b_2 \in B_{\underline{h}(m')}$. Hence,

$$b'_1 \sqcap b'_2 \cong_{m'} \dot{h}_{m'}(b_1) \sqcap \dot{h}_{m'}(b_2).$$

Using 1 of Lemma 8.2.9, $\dot{h}_{m'}(b_1) \sqcap \dot{h}_{m'}(b_2) \cong_{m'} \dot{h}_{m'}(b_1 \sqcap b_2)$. Using transitivity, $\alpha'(\hat{h}(\xi)) \cong_{m'} \dot{h}_{m'}(b_1 \sqcap b_2)$. Therefore, using monotonicity,

$$\ddot{h}_{m'}(\alpha'(\hat{h}(\xi))) \cong_{m'} \ddot{h}_{m'}(\dot{h}_{m'}(b_1 \sqcap b_2)).$$

Using 3 of Lemma 8.2.9 and again transitivity, $\ddot{h}_{m'}(\alpha'(\hat{h}(\xi))) \cong_{m'} b_1 \sqcap b_2$. Using again monotonicity, $\dot{h}_{m'}(\ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))) \cong_{m'} \dot{h}_{m'}(b_1 \sqcap b_2)$. Finally, by transitivity,

$$\dot{h}_{m'}(\ddot{h}_{m'}(\alpha'(\hat{h}(\xi)))) \cong_{m'} \alpha'(\hat{h}(\xi)).$$

2. Proof by induction using Lemma 8.2.9 and Definitions 8.2.1 and 8.2.7.

3. Proof by induction using 1., Lemma 8.2.9, Definitions 8.2.1 and 8.2.7, and noting that if α' is safe for $\hat{h}(\varphi)$ then it is also safe for the subformulas of $\hat{h}(\varphi)$. $\triangleleft$

We now prove that modulated interpretation system morphisms preserve global and local entailments.

Proposition 8.2.16 *Let $h : I \to I'$ be a modulated interpretation system morphism such that $\ddot{h}_{m'}$ is surjective for every $m' \in M'$. Let $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$. In case that $\Gamma \cup \{\varphi\} \cap \Xi \neq \emptyset$ assume, additionally, that $\hat{h} \in S'$. Then*

1. *If $\Gamma \vDash_I^g \varphi$ then $\hat{h}(\Gamma) \vDash_{I'}^g \hat{h}(\varphi)$.*
2. *If $\Gamma \vDash_I^\ell \varphi$ then $\hat{h}(\Gamma) \vDash_{I'}^\ell \hat{h}(\varphi)$.*

Proof. First, observe that if α' is a safe assignment over m' for $\hat{h}(\Phi)$, where $\Phi \subseteq L(\Sigma)$, then the assignment $\underline{h}(\alpha')$ over $\underline{h}(m')$ as defined in Lemma 8.2.15 is safe for Φ. In fact, let $s : \breve{\Sigma} \to \Sigma$ in S and $\breve{\xi} \in \breve{\Xi}$ such that $s(\breve{\xi}) \in \mathrm{Var}(\Phi)$. Thus, $\hat{h}(s(\breve{\xi})) \in \mathrm{Var}(\hat{h}(\Phi))$. Since $\hat{h}$ is a modulated signature morphism and $s \in S$, $\hat{h} \circ s \in S'$. Then

$$\alpha'(\hat{h}(s(\breve{\xi}))) \in B'_{m'}(\hat{h} \circ s)$$

because α' is a safe assignment for $\hat{h}(\Phi)$. By Lemma 8.2.10, exists $b \in B_{\underline{h}(m')}(s)$ such that $\alpha'(\hat{h}(s(\breve{\xi}))) \cong_{m'} \dot{h}_{m'}(b)$. Hence,

$$\ddot{h}_{m'}(\alpha'(\hat{h}(s(\breve{\xi})))) \cong_{\underline{h}(m')} \ddot{h}_{m'}(\dot{h}_{m'}(b))$$

and we conclude $\ddot{h}_{m'}(\alpha'(\hat{h}(s(\breve{\xi})))) \cong_{\underline{h}(m')} b$, that is, $\underline{h}(\alpha')(s(\breve{\xi})) \cong_{\underline{h}(m')} b$, using the properties of $\ddot{h}_{m'}$, transitivity and Lemma 8.2.9. Thus,

$$\underline{h}(\alpha')(s(\breve{\xi})) \in B_{\underline{h}(m')}(s).$$

Therefore $\underline{h}(\alpha')$ over $\underline{h}(m')$ is safe for Φ. Note also that, if $\underline{h}(m')$ is safe for Φ then $\underline{h}(m')$ is safe for each $\varphi \in \Phi$.

1. Let m' be in M' and α' be an assignment over m' safe for $\hat{h}(\Gamma \cup \{\varphi\})$. Assume that $\Gamma \vDash_I^g \varphi$. Then there exists a finite subset Ψ of Γ such that, for each assignment α over m in M safe for $\Psi \cup \{\varphi\}$,

$$[\![\varphi]\!]_m^\alpha \cong_m \top$$

whenever $[\![\psi]\!]_m^\alpha \cong_m \top$ for every $\psi \in \Psi$. Since α' is also safe for $\hat{h}(\Psi \cup \{\varphi\})$, $\underline{h}(\alpha')$ is safe for the set $\Psi \cup \{\varphi\}$. Thus,

$$[\![\varphi]\!]_{\underline{h}(m')}^{\underline{h}(\alpha')} \cong_{\underline{h}(m')} \top$$

whenever $[\![\psi]\!]_{\underline{h}(m')}^{\underline{h}(\alpha')} \cong_{\underline{h}(m')} \top$ for every $\psi \in \Psi$. Consider the finite set $\hat{h}(\Psi) \subseteq \hat{h}(\Gamma)$ and assume that $[\![\hat{h}(\psi)]\!]_{m'}^{\alpha'} \cong_{m'} \top$ for every $\psi \in \Psi$. Using Lemma 8.2.15 and transitivity, $\dot{h}_{m'}([\![\psi]\!]_{\underline{h}(m')}^{\underline{h}(\alpha')}) \cong_{m'} \top$ for every $\psi \in \Psi$. Therefore,

$$\ddot{h}_{m'}(\dot{h}_{m'}([\![\psi]\!]_{\underline{h}(m')}^{\underline{h}(\alpha')})) \cong_{m'} \ddot{h}_{\underline{h}(m')}(\top)$$

for every $\psi \in \Psi$. Using transitivity, Lemma 8.2.9 and recalling that $\ddot{h}$ preserves meets, $[\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')} \cong_{m'} \top$ for every $\psi \in \Psi$. Therefore,

$$[\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')} \cong_{\underline{h}(m')} \top$$

and, by monotonicity,

$$\dot{h}_{m'}([\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}) \cong_{m'} \dot{h}_{m'}(\top).$$

Finally, using transitivity and Lemmas 8.2.9 and 8.2.15, $[\![\hat{h}(\varphi)]\!]^{\alpha'}_{m'} \cong_{m'} \top$. Hence, $\hat{h}(\Gamma) \vDash^{g}_{I'} \hat{h}(\varphi)$.

2. Let m' be in M' and let α' be an assignment over m' safe for $\hat{h}(\Gamma \cup \{\varphi\})$. Assuming that $\Gamma \vDash^{\ell}_{I} \varphi$ then, using Proposition 8.2.14, there exists a finite subset Ψ of Γ such that

$$\sqcap\{[\![\psi]\!]^{\alpha}_{m} : \psi \in \Psi\} \leq_{m} [\![\varphi]\!]^{\alpha}_{m}$$

for every model m in M and assignment α over m safe for $\Psi \cup \{\varphi\}$. As above, $\underline{h}(\alpha')$ is safe for $\Psi \cup \{\varphi\}$. Using Lemma 8.2.15,

$$\sqcap\{[\![\hat{h}(\psi)]\!]^{\alpha'}_{m'} : \psi \in \Psi\} \cong_{m'} \sqcap\{\dot{h}_{m'}([\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}) : \psi \in \Psi\}$$

and, using Lemma 8.2.9,

$$\dot{h}_{m'}(\sqcap\{[\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')} : \psi \in \Psi\}) \cong_{m'} \sqcap\{\dot{h}_{m'}([\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}) : \psi \in \Psi\}.$$

Since

$$\sqcap\{[\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')} : \psi \in \Psi\} \leq_{\underline{h}(m')} [\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}$$

by monotonicity,

$$\dot{h}_{m'}(\sqcap\{[\![\psi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')} : \psi \in \Psi\}) \leq_{m'} \dot{h}_{m'}([\![\varphi]\!]^{\underline{h}(\alpha')}_{\underline{h}(m')}).$$

Finally, using transitivity and Lemma 8.2.15,

$$\sqcap\{[\![\hat{h}(\psi)]\!]^{\alpha'}_{m'} : \psi \in \Psi\} \leq_{m'} [\![\hat{h}(\varphi)]\!]^{\alpha'}_{m'}.$$

Using Proposition 8.2.14, we conclude that $\hat{h}(\Gamma) \vDash^{\ell}_{I'} \hat{h}(\varphi)$. ◁

The modulated fibring mechanism can be now described. The idea is that each model in the modulated fibring of I' and I'' will be a pair of models $\langle m', m''\rangle$ where m' is a model of I' and m'' is a model of I''. The truth values in the algebra of $\langle m', m''\rangle$ should be the (disjoint) union of the truth values in the algebras of m' and m''. However, for denotations of formulas, we need some relationship between the truth values of m' and m'' for every m' and m''. Such a relationship is established by a bridge. As we will see below, bridges modulate the fibring mechanism.

A bridge between two interpretation systems is defined as an appropriated diagram in the category **mInt**.

Definition 8.2.17 A *bridge* between interpretation systems I' and I'' is a diagram

$$\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I'' \rangle$$

in **mInt** such that $\hat{f}'$ and $\hat{f}''$ are injective, $\dot{f}'_{m'}$, $\dot{f}''_{m''}$ are also injective and $\ddot{f}'_{m'}$ and $\ddot{f}''_{m''}$ are surjective for every $m' \in M'$ and $m'' \in M''$. ∇

A bridge between interpretation systems I' and I'' (see Figure 8.6) consists of two interpretation system morphisms f' and f'' from an interpretation system $\breve{I}$ respectively to I' and I''.

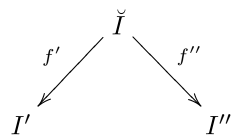


Figure 8.6: Bridge of interpretation systems

The covariant components of f' and f'' are injective. The contravariant components $\ddot{f}'_{m'}$ and $\ddot{f}''_{m''}$ are surjective.

As seen in the previous chapters, the modulated fibring of two modulated interpretation systems corresponds to a pushout in the category **mInt** of a bridge between these modulated interpretation systems.

To make the presentation simpler, before defining modulated fibring we introduce the auxiliary category **poFam** (of families of suitable pre-orders) and two auxiliary functors.

Prop/Definition 8.2.18 *The category* ***poFam*** *is such that*

- *objects are families* $P = \{\langle P_j, \leq_j \rangle\}_{j \in J}$ *of pre-orders with finite meets;*
- *morphisms* $h : \{\langle P_j, \leq_j \rangle\}_{j \in J} \to \{\langle P'_{j'}, \leq'_{j'} \rangle\}_{j' \in J'}$ *are such that* $h = \langle \underline{h}, \dot{h} \rangle$ *where* $\underline{h} : J' \to J$ *is a map and* $\dot{h} = \{\dot{h}_{j'} : P_{\underline{h}(j')} \to P'_{j'}\}_{j' \in J'}$ *is a family of monotonic maps.*

The category ***poFam*** *has pushouts.*

Proof. Let $\beta = \langle \langle \underline{f}', \dot{f}' \rangle : \breve{P} \to P', \langle \underline{f}'', \dot{f}'' \rangle : \breve{P} \to P'' \rangle$.

(i) Consider

$$J = \{\langle j', j'' \rangle : \underline{f}'(j') = \underline{f}''(j''), j' \in J', j'' \in J''\}$$

and $\underline{g}' : J \to J'$, $\underline{g}'' : J \to J''$ the corresponding projections.

(ii) For each $j \in J$, let

$$\langle \langle P_j, \leq_j \rangle, \dot{g}'_j, \dot{g}''_j \rangle$$

be a pushout of $\dot{f}'_{\underline{g}'(j)}$ and $\dot{f}''_{\underline{g}''(j)}$ in the category of pre-orders with finite meets.

(iii) Finally, consider $P = \{\langle P_j, \leq_j \rangle\}_{j \in J}$ and $g' : P' \to P$, $g'' : P'' \to P$ where

$$g' = \langle \underline{g}', \dot{g}' \rangle,\ g'' = \langle \underline{g}'', \dot{g}'' \rangle \text{ with } \dot{g}' = \{\dot{g}'_j\}_{j \in J} \text{ and } \dot{g}'' = \{\dot{g}''_j\}_{j \in J}.$$

Then $\langle P, g', g'' \rangle$ is a pushout of β in **poFam**. ◁

The auxiliary functors are the following. The functor $Sg : \mathbf{mInt} \to \mathbf{mSig}$ (see Fig 8.7) is such that $Sg(I) = \Sigma$ and $Sg(h) = \hat{h}$. Given a diagram τ in **mInt**, $Sg(\tau)$ is the diagram in **mSig** we obtain by replacing each object and morphism in τ by the corresponding image given by Sg.

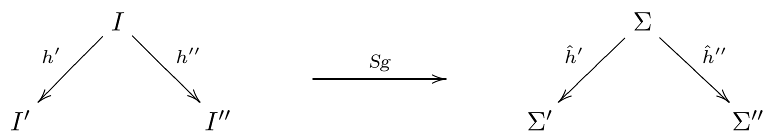


Figure 8.7: Forgetful functor between interpretation systems and signatures

The functor $poF : \mathbf{mInt} \to \mathbf{poFam}$ (see Figure 8.8) is such that $poF(I) = \{\langle B_m, \leq_m \rangle\}_{m \in M}$ and $poF(h) = \langle \underline{h}, \dot{h} \rangle$. Given a diagram τ in poF, $poF(\tau)$ is the diagram in **poFam** we obtain by replacing each object and morphism in τ by the corresponding image given by poF.

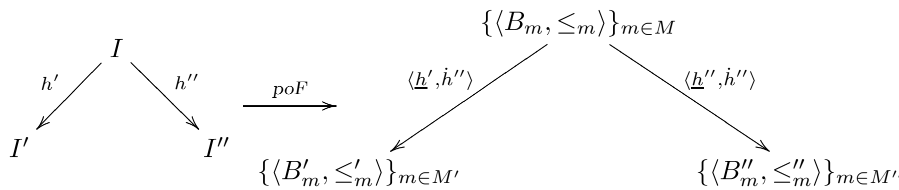


Figure 8.8: Functor between interpretation systems and truth-value algebras

We now define modulated fibring of modulated interpretation systems as a pushout of a bridge between them.

Prop/Definition 8.2.19 *The* modulated fibring *of the modulated interpretation systems I' and I'' by a bridge $\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I'' \rangle$ is a pushout of β in **mInt**, if the pushout of β exists.*

Proof. Consider the bridge $\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I'' \rangle$. Recall that

$$f' = \langle \hat{f}', \underline{f}', \dot{f}', \ddot{f}' \rangle \text{ and } f'' = \langle \hat{f}'', \underline{f}'', \dot{f}'', \ddot{f}'' \rangle$$

where $\dot{f}' = \{\dot{f}'_{m'} : \breve{B}_{\underline{f}'(m')} \to B'_{m'}\}_{m' \in M'}$ and $\dot{f}'' = \{\dot{f}''_{m''} : \breve{B}_{\underline{f}''(m'')} \to B''_{m''}\}_{m'' \in M''}$. The pushout

$$\langle g' : I' \to I, g'' : I'' \to I \rangle$$

of β is defined as follows (see Figure 8.9):

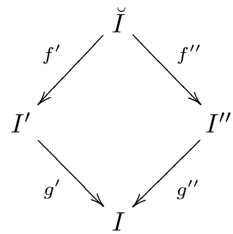


Figure 8.9: Pushout of a bridge of interpretation systems

- $\langle \hat{g}' : Sg(I') \to \Sigma, \hat{g}'' : Sg(I'') \to \Sigma\rangle$ is a pushout in **mSig** of $Sg(\beta)$;
- the pair

$$\langle \langle \underline{g}', \dot{g}'\rangle \colon poF(I') \to \{\langle B_m, \leq_m\rangle\}_{m\in M}$$
$$\langle \underline{g}'', \dot{g}''\rangle \colon poF(I'') \to \{\langle B_m, \leq_m\rangle\}_{m\in M}\rangle$$

  is a pushout in **poFam** of $poF(\beta)$ where $M = \{\langle m', m''\rangle : \underline{f}'(m') = \underline{f}''(m'')\}$ (note that, for each $m = \langle m', m''\rangle \in M$,

$$\langle \dot{g}'_m : B'_{m'} \to B_m, \dot{g}''_m : B''_{m''} \to B_m\rangle$$

  is a pushout of $\langle \dot{f}'_{m'} : B_{\underline{f}'(m')} \to B'_{m'}, \dot{f}''_{m''} : B_{\underline{f}''(m'')} \to B''_{m''}\rangle$);

- $\ddot{g}' = \{\ddot{g}'_{\langle m',m''\rangle} : B_{\langle m',m''\rangle} \to B'_{m'}\}_{\langle m',m''\rangle\in M}$ where, for each $\langle m', m''\rangle \in M$,

$$\ddot{g}'_{\langle m',m''\rangle}(b) = \begin{cases} b' & \text{if } b = \dot{g}'_{\langle m',m''\rangle}(b') \\ \dot{f}'_{m'}(\ddot{f}''_{m''}(b'')) & \text{if } b = \dot{g}''_{\langle m',m''\rangle}(b'') \end{cases}$$

  and

$$\ddot{g}'_{\langle m',m''\rangle}(\dot{g}'_{\langle m',m''\rangle}(b') \sqcap_{\langle m',m''\rangle} \dot{g}''_{\langle m',m''\rangle}(b'')) = \ddot{g}'_{\langle m',m''\rangle}(\dot{g}'_{\langle m',m''\rangle}(b')) \sqcap'_{m'} \ddot{g}'_{\langle m',m''\rangle}(\dot{g}''_{\langle m',m''\rangle}(b''));$$

- $\ddot{g}'' = \{\ddot{g}'_{\langle m',m''\rangle} : B_{\langle m',m''\rangle} \to B''_{m''}\}_{\langle m',m''\rangle\in M}$ is defined in a similar way;
- $I = \langle \Sigma, M, A\rangle$ where $A(\langle m', m''\rangle) = \langle B_{\langle m',m''\rangle}, \leq_{\langle m',m''\rangle}, \nu_{\langle m',m''\rangle}\rangle$, for each $\langle m', m''\rangle \in M$, with
  - $\nu_{\langle m',m''\rangle}(\hat{g}'(c'))(\vec{b}) = \dot{g}'_{\langle m',m''\rangle}(\nu'_{m'}(c')(\ddot{g}'_{\langle m',m''\rangle}(\vec{b})))$;
  - $\nu_{\langle m',m''\rangle}(\hat{g}''(c''))(\vec{b}) = \dot{g}''_{\langle m',m''\rangle}(\nu''_{m''}(c'')(\ddot{g}''_{\langle m',m''\rangle}(\vec{b})))$.

We have to check that

$$\langle I, g', g'' \rangle$$

is a pushout in **mInt** of f' and f''. For this purpose we consider $m' \in M'$ and $m'' \in M''$ and for the sake of simplification will omit the subscripts involving both m' and m''. Moreover we will consider that

$$\underline{f'}(m') = \underline{f''}(m'') = \breve{m}.$$

We just check the properties for $\ddot{g}'$, since the case of $\ddot{g}''$ is similar. Note that $\dot{g}'$ and $\dot{g}'$ are injective and $\ddot{g}'$ and $\ddot{g}'$ are surjective.

1. We now prove that $\ddot{g}'$ is a monotonic map.

It is important at this point to recognize that $\leq \subseteq B^2$ can be defined as a least fixed point. Let D_0 be the subset of B^2 that includes:

- $\dot{g}'(\leq')$ and $\dot{g}''(\leq'')$;
- the pairs $\dot{g}'(b') \sqcap \dot{g}''(b'') \leq \dot{g}'(b')$ for every b' and b'';
- the pairs $\dot{g}'(b') \sqcap \dot{g}''(b'') \leq \dot{g}''(b'')$ for every b' and b'';
- $b \leq \dot{g}'(b') \sqcap \dot{g}''(b'')$ whenever $b \leq \dot{g}'(b')$, $b \leq \dot{g}''(b'')$ and b is $\dot{g}'(\dot{f}'(\breve{b}))$;
- $\dot{g}'(b'_1) \sqcap \dot{g}''(b''_1) \leq \dot{g}'(b'_2) \sqcap \dot{g}''(b''_2)$ whenever $\dot{g}'(b'_1) \leq \dot{g}'(b'_2)$ and $\dot{g}''(b''_1) \leq \dot{g}''(b''_2)$.

Consider the map $\Delta : \wp B^2 \to \wp B^2$ such that $\Delta(D)$ is the one-step transitive closure. Therefore, Δ is extensive and monotonic. Then by the Tarski fixed point theorem there is a least fixed point containing D_0. Call it $lfp(\Delta, D_0)$. We can consider the characterization of the least fixed point in terms of ordinal powers of Δ (useful for proving properties). Consider the following sequence:

- $\Delta^0(D_0) = D_0$;
- $\Delta^{\mu+1}(D_0) = \Delta(\Delta^{\mu}(D_0))$;
- $\Delta^{\mu}(D_0) = \bigcup_{\mu'<\mu}(\Delta^{\mu'}(D_0))$ if a is limit ordinal.

Then for every ordinal μ containing D_0 we have

$$\Delta^{\mu}(D_0) \leq lfp(\Delta, D_0).$$

We prove that

$$\ddot{g}'(b_1) \leq' \ddot{g}'(b_2)$$

whenever $b_1 \leq b_2 \in \Delta^{\mu}(D_0)$ by induction.

Base: $\mu = 0$.

(i) Assume that b_1 and b_2 are either $\dot{g}'(b'_1)$ and $\dot{g}'(b'_2)$ for some $b'_1, b'_2 \in B'$ or $\dot{g}''(b''_1)$ and $\dot{g}''(b''_2)$ for some $b''_1, b''_2 \in B''$. Then

$$\ddot{g}'(b_1) \leq' \ddot{g}'(b_2)$$

by definition of $\leq$, the fact that $\ddot{g}'$ and $\ddot{g}''$ are surjective and the monotonicity of $\dot{f}'', \ddot{f}''$.

(ii) Assume that b_1 is $\dot{g}'(b') \sqcap \dot{g}''(b'')$ and b_2 is $\dot{g}'(b')$. Then $\ddot{g}'(b_2) = b'$ and $\ddot{g}'(b_1)$ is $b' \sqcap' \dot{f}'(\ddot{f}''(b''))$ and so

$$b' \sqcap' \dot{f}'(\ddot{f}''(b'')) \leq' b'.$$

(iii) Assume that b_1 is $\dot{g}'(\dot{f}'(\breve{b})) = \dot{g}''(\dot{f}''(\breve{b}))$ and b_2 is $\dot{g}'(b') \sqcap \dot{g}''(b'')$ with $\dot{f}'(\breve{b}) \leq' b'$ and $\dot{f}''(\breve{b}) \leq'' b''$ (therefore $\breve{b} \breve{\leq} \ddot{f}''(b'')$). Then

$$\ddot{g}'(\dot{g}'(\dot{f}'(\breve{b}))) \cong' \dot{f}'(\breve{b})$$

and

$$\ddot{g}'(\dot{g}''(b'')) \cong' \dot{f}'(\ddot{f}''(b'')).$$

Hence, $\dot{f}'(\breve{b}) \leq' b'$ and $\dot{f}'(\breve{b}) \leq' \dot{f}'(\ddot{f}''(b''))$.

(iv) Assume that

$$b_1 \text{ is } \dot{g}'(b_1') \sqcap \dot{g}''(b_1'') \text{ and } b_2 \text{ is } \dot{g}'(b_2') \sqcap \dot{g}''(b_2'')$$

with $\dot{g}'(b_1') \leq \dot{g}'(b_2')$ and $\dot{g}''(b_1'') \leq \dot{g}''(b_2'')$. So $b_1' \leq' b_2'$, $b_1'' \leq'' b_2''$ and $\dot{f}'(\ddot{f}''(b_1'')) \leq' \dot{f}'(\ddot{f}''(b_2''))$. Then

$$\ddot{g}'(\dot{g}'(b_1')) \leq' \ddot{g}'(\dot{g}'(b_2')) \text{ and } \ddot{g}'(\dot{g}''(b_1'')) \leq' \ddot{g}'(\dot{g}''(b_2'')).$$

Hence

$$\ddot{g}'(\dot{g}'(b_1')) \sqcap' \ddot{g}'(\dot{g}''(b_1'')) \leq' \ddot{g}'(\dot{g}'(b_2')) \sqcap' \ddot{g}'(\dot{g}''(b_2'')).$$

Step:

(i) Case $\mu = \epsilon + 1$. Let b be such that $b_1 \leq b$ and $b \leq b_2 \in D_\epsilon$. By the induction hypothesis, it follows that

$$\ddot{g}'(b_1) \leq' \ddot{g}'(b) \text{ and } \ddot{g}'(b) \leq' \ddot{g}'(b_2)$$

and so, by transitivity of $\leq'$, we have $\ddot{g}'(b_1) \leq' \ddot{g}'(b_2)$.

(ii) Case μ is a limit ordinal. Straightforward.

2. Preservation of meets by $\ddot{g}'$ and $\ddot{g}''$: Straightforward.

3. We now prove that $\ddot{f}'(\ddot{g}'(b)) \cong \ddot{f}''(\ddot{g}''(b))$.
Let b be $\dot{g}'(b')$. Therefore,

$$\ddot{f}'(\ddot{g}'(\dot{g}'(b'))) \cong \ddot{f}'(b') \text{ and } \ddot{f}''(\ddot{g}''(\dot{g}'(b'))) \cong \ddot{f}''(\dot{f}''(\ddot{f}'(b')))$$

and so $\ddot{f}''(\dot{f}''(\ddot{f}'(b'))) \cong \ddot{f}'(b')$, since $\ddot{f}''$ is surjective. The other cases follow straightforwardly.

4. We have that

$$\begin{aligned}\nu(\hat{g}'(\hat{f}'(\breve{c})))(\vec{b}) &\cong \dot{g}'(\nu'(\hat{f}'(\breve{c}))(\ddot{g}'(\vec{b})))\\ &\cong \dot{g}'(\dot{f}'(\breve{\nu}(\breve{c})(\ddot{f}'(\ddot{g}'(\vec{b})))))\\ &\cong \dot{g}''(\dot{f}''(\breve{\nu}(\breve{c})(\ddot{f}''(\ddot{g}''(\vec{b})))))\\ &\cong \dot{g}''(\nu''(\hat{f}''(\breve{c}))(\ddot{g}''(\vec{b})))\\ &\cong \nu(\hat{g}''(\hat{f}''(\breve{c})))(\vec{b}).\end{aligned}$$

5. We now show that $\ddot{g}'$ is left adjoint of $\dot{g}'$ ($\ddot{g}''$ is left adjoint of $\dot{g}''$).

(i) $b \leq \dot{g}'(\ddot{g}'(b))$. Consider the case of b being $\dot{g}''(b'')$. We have

$$b'' \leq'' \dot{f}''(\ddot{f}''(b''))$$

and therefore $\dot{g}''(b'') \leq \dot{g}''(\dot{f}''(\ddot{f}''(b'')))$. Hence,

$$\dot{g}''(b'') \leq \dot{g}'(\dot{f}'(\ddot{f}''(b''))) \text{ and } \dot{g}''(b'') \leq \dot{g}'(\ddot{g}'(\dot{g}''(b''))).$$

(ii) $\ddot{g}'(\dot{g}'(b')) \leq b'$. Straightforward.

6. Universal property.

Let

$$h' : I' \to I''' \text{ and } h'' : I'' \to I'''$$

be interpretation system morphisms such that $h' \circ f' = h'' \circ f''$.

Existence: Consider

$$h = \langle \hat{h}, \underline{h}, \dot{h}, \ddot{h}_{m'''} \rangle$$

such that

- $\hat{h}$ is the unique morphism in **mSig** such that $\hat{h} \circ \hat{g}' = \hat{h}'$ and $\hat{h} \circ \hat{g}'' = \hat{h}''$;
- $\underline{h} = \langle \underline{h}', \underline{h}'' \rangle$;
- $\dot{h}$ is the unique morphism in **poFam** such that $\dot{h} \circ \dot{g}' = \dot{h}'$ and $\dot{h} \circ \dot{g}'' = \dot{h}''$;
- $\ddot{h}_{m'''}(b''') = \dot{g}'_{\underline{h}(m''')}(\ddot{h}'_{m'''}(b''')) \sqcap \dot{g}''_{\underline{h}(m''')}(\ddot{h}''_{m'''}(b'''))$.

Then,

$$\begin{aligned}\ddot{g}'(\ddot{h}(b''')) &\cong \ddot{g}'(\dot{g}'(\ddot{h}'(b'''))) \sqcap \ddot{g}'(\dot{g}''(\ddot{h}''(b''')))\\ &\cong \ddot{h}'(b''') \sqcap \dot{f}'(\ddot{f}''(\ddot{h}''(b''')))\\ &\cong \ddot{h}'(b''') \sqcap \dot{f}'(\ddot{f}'(\ddot{h}'(b''')))\\ &\cong \ddot{h}'(b''').\end{aligned}$$

We can also conclude that $\ddot{h}$ is monotonic and preserves finite meets and that $\ddot{h}$ is left adjoint to $\dot{h}$. Finally,

$$\begin{aligned}\nu'''(\hat{h}(\hat{g}'(c')))(\vec{b}''') &\cong \nu'''(\hat{h}'(c'))(\vec{b}''')\\ &\cong \dot{h}'(\nu'(c')(\ddot{h}'(\vec{b}''')))\\ &\cong \dot{h}(\dot{g}'(\nu'(c')(\ddot{g}'(\ddot{h}(\vec{b}''')))))\\ &\cong \dot{h}(\nu(\hat{g}'(c'))(\ddot{h}(\vec{b}'''))).\end{aligned}$$

Uniqueness: Assume that $k : I \to I'''$ is a morphism such that $k \circ g' = h'$ and $k \circ g'' = h''$. We want to show that $k = h$ that is $\ddot{k} = \ddot{h}$. We start by showing that

$$\ddot{k}(b''') \cong \dot{g}'(\ddot{g}'(\ddot{k}(b'''))) \sqcap \dot{g}''(\ddot{g}''(\ddot{k}(b'''))).$$

Note that $\ddot{k}(b''') \leq \dot{g}''(\ddot{g}''(\ddot{k}(b''')))$, by the left adjoint condition. Assume that $\ddot{k}(b''') = \dot{g}'(b')$. Using the definition of $\ddot{g}''$, $b' = \ddot{g}'(\dot{g}'(b'))$. Thus,

$$\begin{aligned} \ddot{k}(b''') &= \dot{g}'(b') \\ &= \dot{g}'(\ddot{g}'(\dot{g}'(b'))) \\ &= \dot{g}'(\ddot{g}'(\ddot{k}(b'''))). \end{aligned}$$

Therefore, $\ddot{k}(b''') \leq \dot{g}'(\ddot{g}'(\ddot{k}(b'''))) \sqcap \dot{g}''(\ddot{g}''(\ddot{k}(b''')))$. Moreover,

$$\dot{g}'(\ddot{g}'(\ddot{k}(b'''))) \sqcap \dot{g}''(\ddot{g}''(\ddot{k}(b'''))) \leq \ddot{k}(b''')$$

since $\ddot{k}(b''') = \dot{g}'(\ddot{g}'(\ddot{k}(b''')))$. Thus,

$$\ddot{k}(b''') \cong \dot{g}'(\ddot{g}'(\ddot{k}(b'''))) \sqcap \dot{g}''(\ddot{g}''(\ddot{k}(b'''))).$$

Since $\ddot{g}'(\ddot{k}(b''')) = \ddot{h}'(b''')$ and $\ddot{g}''(\ddot{k}(b''')) = \ddot{h}''(b''')$, we conclude that $k = h$. $\triangleleft$

Observe that it may be the case that there is no pushout of a given bridge β. We illustrate this situation in Example 8.2.20.

Example 8.2.20 Consider the bridge $\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I'' \rangle$ presented in Figure 8.10: where, in particular, there is $\breve{m} \in \breve{M}$ such that $\underline{f}'(m') = \underline{f}''(m'') = \breve{m}$,

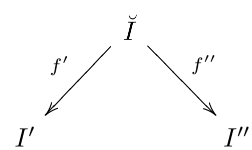


Figure 8.10: Example of a bridge of interpretation systems

for some $m' \in M'$, $m'' \in M''$ such that

- $\breve{B}_{\breve{m}} = \{\breve{b}_1, \breve{b}_2\}$;
- $B'_{m'} = \{b'_1, b'_2\}$ and
  - $\dot{f}'_{m'}(\breve{b}_1) = b'_1, \quad \dot{f}'_{m'}(\breve{b}_2) = b'_2$;
  - $\ddot{f}'_{m'}(b'_1) = \breve{b}_1, \quad \ddot{f}''_{m'}(b'_2) = \breve{b}_2$;
- $B''_{m''} = \{b''_1, b''_2\}$ and

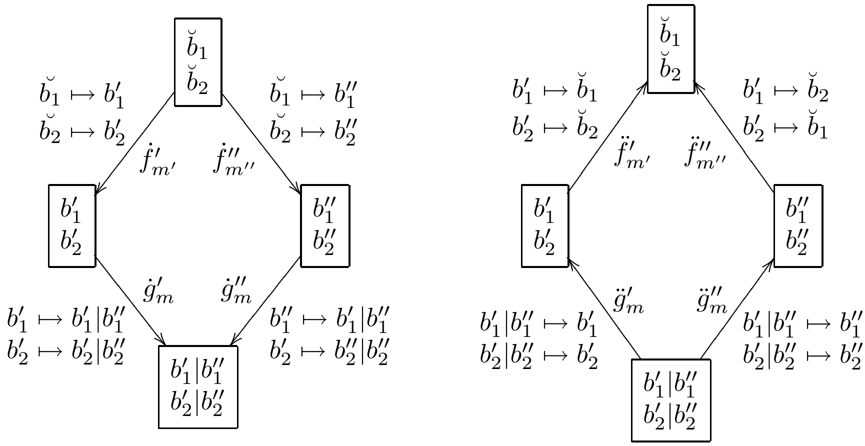


Figure 8.11: Example of a pushout of interpretation systems

– $\dot{f}''_{m''}(\breve{b}_1) = b''_1, \quad \dot{f}''_{m''}(\breve{b}_2) = b'_2;$

– $\ddot{f}''_{m''}(b''_1) = \breve{b}_2, \quad \ddot{f}''_{m''}(b''_2) = \breve{b}_1;$

and the pre-orders involved are isomorphisms.

Let us follow Prop/Definition 8.2.19 to get a pushout $\langle g' : I' \to I, g'' : I'' \to I\rangle$ of β (see Figure 8.11). In particular,

$$\langle \dot{g}'_m : B'_{m'} \to B_m, \dot{g}''_m : B'_{m'} \to B_m\rangle$$

is a pushout of $\langle \dot{f}'_{m'} : \breve{B}_{\breve{m}} \to B'_{m'}, f''_{m''} : \breve{B}_{\breve{m}} \to B''_{m''}\rangle$, where $m = \langle m', m''\rangle$. Hence, we can consider $B_m = \{b'_1|b''_1, b'_2|b''_2\}$ and $\dot{g}'_m(b'_1) = \dot{g}''_m(b''_1) = b'_1|b''_1$ and $\dot{g}'_m(b'_2) = \dot{g}''_m(b''_2) = b'_2|b''_2$.

We now consider the maps

$$\ddot{g}'_m : B_m \to B'_{m'} \text{ and } \ddot{g}''_m : B_m \to B'_{m'}.$$

Following Prop/Definition 8.2.19,

$$\ddot{g}'_m(b'_1|b''_1) = \ddot{g}'_m(\dot{g}'_m(b'_1)) = b'_1,\ \ddot{g}'_m(b'_2|b''_2) = \ddot{g}'_m(\dot{g}'_m(b'_2)) = b'_2,$$
$$\ddot{g}''_m(b'_1|b''_1) = \ddot{g}''_m(\dot{g}''_m(b''_1)) = b''_1$$

and $\ddot{g}''_m(b'_2|b''_2) = \ddot{g}''_m(\dot{g}''_m(b'_2)) = b''_2$. But, in particular, since $b'_1|b''_1 = \dot{g}''_m(b''_1)$, the equality $\ddot{g}'_m(b'_1|b''_1) = \dot{f}_{m'}(\ddot{f}_{m''}(b''1))$ should also hold. However,

$$\dot{f}_{m'}(\ddot{f}_{m''}(b''1)) = b'_2 \neq b'_1.$$

As a consequence, the pushout of β does not exist. ∇

Next, we present some examples of modulated fibring illustrating how the collapse can be avoided. We start by a description of the most common collapse and then give a result stating how the bridge can be chosen to avoid the collapse when no constructors are shared.

Definition 8.2.21 In the modulated fibring $\langle g' : I' \to I, g'' : I'' \to I \rangle$ of I' and I'' by a bridge β, I'' *collapses* to I' whenever there is a bijection $j_k : C''_k \to C'_k$ for all $k \in \mathbb{N}$ such that, for all $\Gamma' \cup \{\varphi'\} \subseteq L(\Sigma')$,

- $\hat{g}'(\Gamma') \vDash^g_I \hat{g}'(\varphi')$ if and only if $\hat{g}''(j^{-1}(\Gamma')) \vDash^g_I \hat{g}''(j^{-1}(\varphi'))$ if and only if $\Gamma' \vDash^g_{I'} \varphi'$;
- $\hat{g}'(\Gamma') \vDash^\ell_I \hat{g}'(\varphi')$ if and only if $\hat{g}''(j^{-1}(\Gamma')) \vDash^\ell_I \hat{g}''(j^{-1}(\varphi'))$ if and only if $\Gamma' \vDash^\ell_{I'} \varphi'$;
- there is a set $\Gamma'' \subseteq L(\Sigma'')$ and a formula $\varphi'' \in L(\Sigma'')$ such that $\Gamma'' \nvDash^g_{I''} \varphi''$ and $\hat{g}''(\Gamma'') \vDash^g_I \hat{g}''(\varphi'')$ or there is a set $\Gamma'' \subseteq L(\Sigma'')$ and a formula $\varphi'' \in L(\Sigma'')$ such that $\Gamma'' \nvDash^\ell_{I''} \varphi''$ and $\hat{g}''(\Gamma'') \vDash^\ell_I \hat{g}''(\varphi'')$. ∇

The basic idea about a collapse is that for each connective of one of the original logics there is a connective in the other original logics that behaves in the same way and vice versa. Hence we require that there is a bijection indicating which connectives correspond to the others. We assume that if c' collapses to c'' then c'' also collapses into c'. For instance when we say that classical logic collapse to intuitionistic logic we are thinking about the bijection where the implications, the negations, the conjunctions and the disjunctions are the same.

The same behaviour means that the semantic consequence in one of the original logic is the same as the semantic consequence in the other original logic providing that we use the bijection for changing the names of the connectives (that is precisely what is required in the first part of the definition). Moreover, we assume that there was a formula that was not entailed in one of the original logic but is entailed in the fibring (that is the content of clause two of the definition). For instance, it is not the case in intuitionistic logic that $\vDash' (\varphi' \vee' (\neg' \varphi'))$ but it is the case when the consider the fibring (not the modulated fibring) of intuitionistic and classical propositional logics.

We now define a specific bridge that leads to a non-collapsing situation whenever there is no sharing of constructors.

In the sequel, we assume that $\mathbf{t}$ (possibly with superscripts) is a constructor of arity 0, designated *verum*, and we assume that its denotation is always $\top$ (again possibly with superscripts). Similarly, we sometimes also assume that $\mathbf{f}$ is a constructor of arity 0, designated *falsum*.

Proposition 8.2.22 *Consider the modulated interpretation systems I', I'' such that*

- $\mathbf{t}' \in C'_0$ *and* $\mathbf{t}'' \in C''_0$;

- *C' and C'' are isomorphic in **Sig** (recall Remark 2.1.10);*
- *$id_{\Sigma'} \in S'$ and $id_{\Sigma''} \in S''$;*

and a bridge $\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I''\rangle$ such that

- *$\breve{C}_0 = \{\breve{\mathbf{t}}\}$ and $\breve{C}_k = \emptyset$ for all $k \neq 0$;*
- *$\breve{\Xi} = \emptyset$ and $\breve{S} = \emptyset$;*
- *$\breve{M} = \{\breve{m}\}$;*
- *$\breve{B}_{\breve{m}} = \{\breve{\top}\}$;*
- *$\dot{f}'_{m'}(\breve{\top}) = \top'_{m'}$, $\dot{f}''_{m''}(\breve{\top}) = \top''_{m''}$ for every $m' \in M'$, $m'' \in M''$.*

Then the modulated fibring $\langle g' : I' \to I, g'' : I'' \to I\rangle$ of I' and I'' by β does not collapse.

Proof. Begin by observing that the mappings $\underline{f'}$, $\underline{f''}$, $\ddot{f}'_{m'}$ and $\ddot{f}''_{m''}$ are constant.

For every model $m'' \in M''$ all the pairs $\langle m', m''\rangle$ with $m' \in M'$ are in the modulated fibring. Therefore

$$\text{if } \Gamma'' \not\vDash^{g}_{I''} \varphi'' \text{ then } \hat{g}''(\Gamma'') \not\vDash^{g}_{I} \hat{g}''(\varphi'')$$

for every Γ'' and φ'' and if $\Gamma'' \not\vDash^{\ell}_{I''} \varphi''$ then $\hat{g}''(\Gamma'') \not\vDash^{\ell}_{I} \hat{g}''(\varphi'')$ for every Γ'' and φ''. $\triangleleft$

Whenever the conditions of Proposition 8.2.22 hold, we say that the modulated interpretation system obtained is the *unconstrained modulated fibring* of I' and I''. Thus, we can use this "universal" bridge for defining the modulated fibring whenever we do not want to share any symbols besides *verum*, which is the case in most situations. Note that in C'_0 and C''_0 we can have propositional symbols. These notion of unconstrained modulated fibring generalizes the notion of unconstrained fibring in a natural way.

Proposition 8.2.22 establishes that the collapsing problem is avoided for all cases of unconstrained modulated fibring, that is when only the *verum* is shared. Since $id_{\Sigma'} \in S'$, $id_{\Sigma''} \in S''$ using Proposition 8.2.16 we guarantee that the entailments of the component logics will be entailments in the modulated fibring. We now sketch this modulated fibring, describing first the relationships between the signatures, then the relationships between the models (see Figure 8.12) and, afterwards, the relationships between the truth values in $\breve{B}_{\breve{m}}$, $B'_{m'}$, $B''_{m''}$ and B_m (see Figure 8.13), for each $m' \in M'$, $m'' \in M''$ and $m = \langle m', m''\rangle \in M' \times M''$.

Note that the requirement $id_{\Sigma'} \in S'$, $id_{\Sigma''} \in S''$ does not change the entailments of I' and I''. This requirement just prepares the modulated interpretation systems for the combination.

We now instantiate Proposition 8.2.22 for several cases. The first example concerns the fibring of classical and intuitionistic logic.

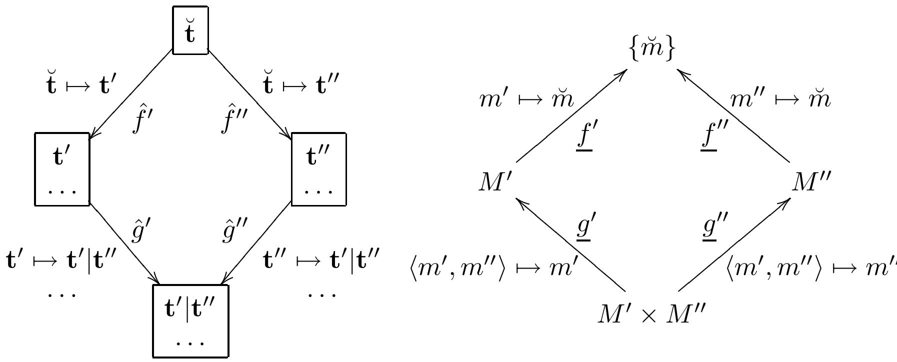


Figure 8.12: Relationship between signatures and models

Example 8.2.23 Let I' and I'' be the modulated interpretation systems for classical logic in Example 8.2.3 and intuitionistic logic in Example 8.2.4. A bridge as the one in Proposition 8.2.22 avoids the collapsing between I' and I''. Intuitionistic logic collapses into classical logic when the formula

$$((\neg(\neg \varphi)) \Leftrightarrow \varphi)$$

becomes valid which is not the case when considering this modulated fibring. Observe that in the modulated fibring, $\dot{g}'(B'_{m'})$ is a Boolean algebra "equivalent" to $B'_{m'}$ and $\dot{g}''(B''_{m''})$ is a Heyting algebra "equivalent" to $B''_{m''}$. ∇

Similarly to Fariñas del Cerro and Herzig's C+J logic as presented in [82], in the modulated fibring of classical logic I' and intuitionistic I'' logic considered above, we have also no problems with the validity of the formula

$$\hat{g}'(\varphi' \Rightarrow' (\psi' \Rightarrow' \varphi'))$$

since, according to our semantics, the formula is only valid for "intuitionistic values". Classical values are converted to intuitionistic value "**t**".

The next example is again an application of Proposition 8.2.22, dealing now with the fibring of classical and Łukasiewicz logics. Moreover it is also very interesting in showing the need for safe assignments.

Example 8.2.24 Let I' and I'' be the interpretation systems for classical logic in Example 8.2.3 and the 3-valued Łukasiewicz logic in Example 8.2.6. As a corollary of Proposition 8.2.22, the modulated fibring with no sharing does not collapse.

In order to understand safe assignments consider the following case. We have

$$\{\xi'_1, (\xi'_1 \Rightarrow' \xi'_2)\} \vDash^{\ell}_{I'} \xi'_2$$

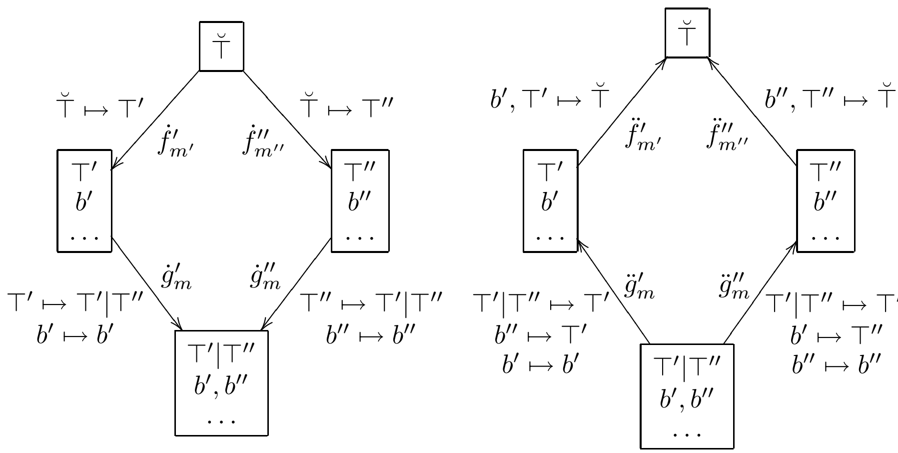


Figure 8.13: Relationship between truth values

for classical logic. In the unconstrained modulated fibring, we do not have

$$\{\hat{g}'(\xi'_1), (\hat{g}'(\xi'_1)\hat{g}'(\Rightarrow')\hat{g}'(\xi'_2))\} \vDash^{\ell}_{I'} \hat{g}'(\xi'_2)$$

if all assignments are possible. Let m' and m'' be such that $B'_{m'} = \{0', 1'\}$ and $B''_{m''} = \{0'', 1/2'', 1''\}$. Then

$$B_{\langle m', m''\rangle} = \{0', 0'', 1/2'', 1\}.$$

Consider an assignment α over $\langle m', m''\rangle$ such that $\alpha(\hat{g}'(\xi'_1)) = 1$ and $\alpha(\hat{g}'(\xi'_2)) = 1/2''$. Then

- $1 \leq [\![\hat{g}'(\xi'_1)]\!]^{\alpha}_{\langle m', m''\rangle}$ since $[\![\hat{g}'(\xi'_1)]\!]^{\alpha}_{\langle m', m''\rangle} = 1$;
- $1 \leq [\![(\hat{g}'(\xi'_1)\hat{g}'(\Rightarrow')\hat{g}'(\xi'_2))]\!]^{\alpha}_{\langle m', m''\rangle}$ since $[\![(\hat{g}'(\xi'_1)\hat{g}'(\Rightarrow')\hat{g}'(\xi'_2))]\!]^{\alpha}_{\langle m', m''\rangle} = 1$;
- but not $1 \leq [\![\hat{g}'(\xi'_2)]\!]^{\alpha}_{\langle m', m''\rangle}$ since $[\![\hat{g}'(\xi'_2)]\!]^{\alpha}_{\langle m', m''\rangle} = 1/2''$. ∇

The following example illustrates several possible combinations of classical logic and Gödel logic through different bridges. In particular we introduce a specific bridge for sharing negation. The motivation for the sharing comes from the fact that the values of $(\neg\,\varphi)$ in 3-valued Gödel logic is always either $\perp$ or $\top$. That is, 1/2 behaves as $\top$, and so negation has a classical flavor.

Example 8.2.25 Let I' and I'' be the interpretation systems for the 3-valued Gödel logic in Example 8.2.5 and classical logic in Example 8.2.3. For classical logic only 2-valued algebras are included. Consider the fibring of classical and Gödel logics modulated by three different bridges

$$\beta = \langle f' : \breve{I} \to I', f'' : \breve{I} \to I''\rangle$$

as follows:

Bridge 1:

- $\breve{I}$ is such that
  - $\breve{M} = \{\breve{m}\}$;
  - $\breve{A}(\breve{m}) = \langle\{\breve{\top}\}, \{\langle\breve{\top}, \breve{\top}\rangle\}, \breve{\nu}\rangle$;
- f' and f'' are such that
  - $\dot{f}'_{m'}(\breve{\top}) = \top'_{m'}$ and $\dot{f}''_{m''}(\breve{\top}) = \top''_{m''}$.

Observe that the mappings $\underline{f}'$, $\underline{f}''$, $\ddot{f}'_{m'}$ and $\ddot{f}''_{m''}$ are constant.

Bridge 2:

- $\breve{I}$ is such that
  - $\breve{M} = \{\breve{m}\}$;
  - $\breve{A}(\breve{m}) = \langle\{\breve{\bot}, \breve{\top}\}, \{\langle\breve{\bot}, \breve{\bot}\rangle, \langle\breve{\bot}, \breve{\top}\rangle, \langle\breve{\top}, \breve{\top}\rangle\}, \breve{\nu}\rangle$;
- f' and f'' are such that
  - $\dot{f}'_{m'}(\breve{\bot}) = \bot'_{m'}$, $\dot{f}'_{m'}(\breve{\top}) = \top'_{m'}$;
  - $\dot{f}''_{m''}(\breve{\bot}) = \bot''_{m''}$ and $\dot{f}''_{m''}(\breve{\top}) = \top''_{m''}$;
  - $\ddot{f}'_{m'}(\bot') = \breve{\bot}$ and $\ddot{f}'_{m'}(b') = \breve{\top}$ for every $b' \neq \bot'_{m'}$;
  - $\ddot{f}''_{m''}(\bot''_{m''}) = \breve{\bot}$ and $\ddot{f}''_{m''}(b'') = \breve{\top}$ for every $b'' \neq \bot''_{m''}$;

Observe that $\underline{f}'$ and $\underline{f}''$ are constant mappings.

Bridge 3:

- $\breve{I}$ is such that
  - $\breve{M} = A'(M')|_{\breve{C}} \cup A''(M'')|_{\breve{C}}$;
  - $\breve{A}$ is the identity map;
- f' and f'' are such that
  - $\underline{f}'(m') = A'(m')|_{\breve{C}}$ and $\underline{f}''(m'') = A''(m'')|_{\breve{C}}$;
  - $\dot{f}'_{m'} = \mathrm{id}_{B'_{m'}}$ and $\dot{f}''_{m''} = \mathrm{id}_{B''_{m''}}$;
  - $\ddot{f}'_{m'} = \mathrm{id}_{\breve{B}_{\underline{f}'(m')}}$ and $\ddot{f}''_{m''} = \mathrm{id}_{\breve{B}_{\underline{f}''(m'')}}$.

Note that bridge 1 is similar to the bridge considered in Proposition 8.2.22.

Bridges 1, 2 and 3 can be used to modulate the fibring when $\breve{C}_0 = \{\breve{\mathbf{t}}\}$ and $\breve{C}_k = \emptyset$, $\breve{\Xi} = \emptyset$ and $\breve{S} = \emptyset$. Then $\breve{\nu}$ is a family of empty maps except for $\breve{\nu}_0$ and $\hat{f}'$ and $\hat{f}''$ are also empty maps except for $k = 0$.

Bridges 2 and 3 can be used to modulate the fibring when $\breve{C}_0 = \{\breve{\mathbf{f}}, \breve{\mathbf{t}}\}$, $\breve{C}_1 = \{\breve{\neg}\}$, $\breve{C}_k = \emptyset$ for every $k \geq 2$, $\breve{\Xi} = \emptyset$, $\breve{S} = \emptyset$, $\breve{\nu}(\breve{\neg})(\breve{\bot}) = \breve{\top}$, $\breve{\nu}(\breve{\neg})(\breve{\top}) = \breve{\bot}$ and $\hat{f}'$ and $\hat{f}''$ are such that $\hat{f}'(\breve{\neg}) = \neg'$ and $\hat{f}''(\breve{\neg}) = \neg''$.

Bridge 3 can be used to modulate the fibring when $\breve{C} = C' = C''$, $\breve{\Xi} = \emptyset$, $\breve{S} = \emptyset$ and $\hat{f}'$ and $\hat{f}''$ are such that $\hat{f}'(\breve{\neg}) = \neg'$, $\hat{f}'(\breve{\wedge}) = \wedge'$, $\hat{f}''(\breve{\neg}) = \neg''$ and $\hat{f}''(\breve{\wedge}) = \wedge''$ (corresponding to the collapse of Gödel logics into classical logics since in the fibring we will only have Boolean algebras). ▽

At the end of this section, we turn our attention to the comparison at the semantic level between modulated fibring and the fibring as presented in [282], and referred in Section 3.4, showing that the latter is a particular case of the former.

Remark 8.2.26 Consider the subcategory **fInt** of **mInt** whose objects are tuples $\langle \Sigma, M, A \rangle$ such that $S = \emptyset$ and the morphisms

$$h : \langle \Sigma, M, A \rangle \to \langle \Sigma', M', A' \rangle$$

are such that $\Xi' = \Xi$, $\dot{h}_{m'} = id_{m'}$ and $\ddot{h}_{m'} = id_{\underline{h}(m')}$ for every $m' \in M'$. The objects and the morphisms of the subcategory **fInt** are the interpretation systems and the morphisms in the fibring as presented in [282], and referred in Chapter 3. The category **fInt** has pushouts that correspond to (unconstrained and constrained) fibring as presented in therein by choosing the following bridge:

- $\breve{C}$ with the shared constructors if any;
- $\breve{M} = A'(M')|_{\breve{C}} \cup A''(M'')|_{\breve{C}}$;
- $\breve{A}$ is the identity map;
- $\underline{f}'(m') = A'(m')|_{\breve{C}}$ and $\underline{f}''(m'') = A''(m'')|_{\breve{C}}$;
- $\dot{f}'_{m'} = \mathrm{id}_{B'_{m'}}$, $\dot{f}''_{m''} = \mathrm{id}_{B''_{m''}}$, $\ddot{f}'_{m'} = \mathrm{id}_{\breve{B}_{\underline{f}'(m')}}$ and $\ddot{f}''_{m''} = \mathrm{id}_{\breve{B}_{\underline{f}''(m'')}}$.

Thus, the class of models M is composed by the pairs $\langle m', m'' \rangle$ that have the same underlying algebra. For instance when considering the fibring of classical and intuitionistic logics the models to be considered in the fibring are those whose underlying algebra is Boolean. Therefore intuitionistic logic collapses into classical logic even if no constructors are shared. ▽

8.3 Modulated Hilbert calculi

We now analyze the deductive counterpart of modulated fibring. As before, the basic deductive notion is the Hilbert calculus. The Hilbert calculi that are relevant herein are Hilbert calculi similar to the ones presented in Section 2.3 of Chapter 2, but where some particular formulas involving the operator & can be derived and including, as expected, modulated signatures. Global rules and local inference rules are also considered. These calculi are called modulated Hilbert calculi. We introduce the notion of morphism between these Hilbert calculi and then, again, modulated fibring appears as a pushout in the category of modulated Hilbert calculi.

The notion of substitution is a delicate one. We will work often with safe substitutions which are a deductive counterpart of safe assignments. This means that instantiation of inference rules is sometimes restricted.

In the sequel, we assume modulated signatures $\Sigma = \langle C, \&, \Xi, S\rangle$, possibly with superscripts. Recall that $gL(\Sigma)$ denotes the set of ground formulas in $L(\Sigma)$. Given $s : \breve{\Sigma} \to \Sigma$, we also use the following notation:

- $L(\Sigma, s)$ is the set of formulas in $L(\Sigma)$ whose main constructor is & or it is in $s(\breve{C})$, that is, formulas $(\varphi_1 \& \varphi_2)$ or $c(\varphi_1, \ldots, \varphi_n)$ where c is a constructor of $s(\breve{C})$;
- $gL(\Sigma, s)$ is the set of all ground formulas in $L(\Sigma, s)$;
- $L(\Sigma, C)$ is the set of formulas in $L(\Sigma)$ whose main constructor is in C;
- $gL(\Sigma, C)$ is the set of all ground formulas in $L(\Sigma, C)$.

Definition 8.3.1 A *substitution* over Σ is a map $\sigma : \Xi \to L(\Sigma)$. A substitution σ is *safe* for a set of formulas $\Gamma \subseteq L(\Sigma)$ whenever

$$\sigma(s(\breve{\xi})) \in L(\Sigma, s)$$

for every $s : \breve{\Sigma} \to \Sigma$ in S and $s(\breve{\xi}) \in \mathrm{Var}(\Gamma)$. ▽

A safe substitution σ for a set of formulas Γ always associates a particular kind of formulas to the schema variables in the range of safe-relevant signature morphisms that occur in formulas of Γ. To such schema variables, images of schema variables of some signature $\breve{\Sigma}$, the safe substitution σ only associates formulas whose main constructor belongs to the signature $\breve{\Sigma}$.

In the sequel, when we refer to a Hilbert calculus $H = \langle C, R_g, R_\ell\rangle$, we assume the Definition 2.3.1.

Definition 8.3.2 A *modulated pre-Hilbert calculus* is a tuple

$$H = \langle \Sigma, R_g, R_\ell\rangle$$

where $H = \langle C, R_g, R_\ell \rangle$ is a Hilbert calculus. The definition of global and local derivations in H follow Definition 2.3.2 but, whenever a rule $\langle \Delta, \delta \rangle$ is used in a derivation, only safe assignments for $\Delta \cup \{\delta\}$ can be considered. ∇

Notations corresponding to global and local derivations are as before. We denote by $\varphi_1 \cong_{H,\Gamma} \varphi_2$ the fact that $\Gamma, \varphi_1 \vdash^\ell_H \varphi_2$ and $\Gamma, \varphi_2 \vdash^\ell_H \varphi_1$. When $\Gamma = \emptyset$, we will omit the reference to the set. We will omit the reference to the calculus whenever it is clear which one we are considering.

Definition 8.3.3 A *modulated Hilbert calculus* is a modulated pre-Hilbert calculus $H = \langle \Sigma, R_g, R_\ell \rangle$ where, for every formulas $\varphi_1, \varphi_2 \in L(\Sigma)$,

- $\{(\varphi_1 \& \varphi_2)\} \vdash^\ell_H \varphi_1$ and $\{(\varphi_1 \& \varphi_2)\} \vdash^\ell_H \varphi_2$;
- $\{\varphi_1, \varphi_2\} \vdash^\ell_H (\varphi_1 \& \varphi_2)$. ∇

The first two conditions are called & *elimination* and the last one is called & *introduction*. These conditions show that in modulated Hilbert calculi the symbol & is a conjunction like operator in what concerns deduction. This operator has a technical role later on, when studying preservation properties. Clearly, & elimination and introduction can be extended to any finite number of formulas (for simplicity we omit the inner parenthesis).

We now present two examples of modulated Hilbert calculi. They correspond to the many-values logics introduced before. In both examples the signature Σ is as follows: $\Sigma = \langle C, \&, \Xi, S \rangle$ where $\mathbf{t}, \mathbf{f} \in C_0$, $C_1 = \{\neg\}$, $C_2 = \{\wedge, \vee, \Rightarrow\}$, $C_k = \emptyset$ for all $k \geq 3$, & is $\wedge$ and $\Xi = \{\xi_i : i \in \mathbb{N}\}$.

Example 8.3.4 Recall Example 2.2.8. A modulated Hilbert calculus

$$H = \langle \Sigma, R_g, R_\ell \rangle$$

for 3-valued Łukasiewicz logic is such that R_g consists, besides the rules of Definition 8.3.3, of the following inference rules

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, ((\neg \xi_1) \Rightarrow (\neg \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, (((\xi_1 \Rightarrow (\neg \xi_1)) \Rightarrow \xi_1) \Rightarrow \xi_1) \rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$;

and $R_\ell = R_g$. ∇

Example 8.3.5 Recall Example 2.2.7. A modulated Hilbert calculus

$$H = \langle \Sigma, R_g, R_\ell \rangle$$

for 3-valued Gödel logic is such that R_g consists, besides the rules of Definition 8.3.3, of the following inference rules

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow (\xi_1 \wedge \xi_2))) \rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_1) \rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_2) \rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_1 \vee \xi_2)) \rangle$;
- $\langle \emptyset, (\xi_2 \Rightarrow (\xi_1 \vee \xi_2)) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_3) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow ((\xi_1 \vee \xi_2) \Rightarrow \xi_3))) \rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_2) \Rightarrow ((\xi_1 \Rightarrow (\neg \xi_2)) \Rightarrow (\neg \xi_1))) \rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow ((\neg \xi_1) \Rightarrow \xi_2)) \rangle$;
- $\langle \emptyset, (((\neg \xi_1) \Rightarrow \xi_2) \Rightarrow (((\xi_2 \Rightarrow \xi_1) \Rightarrow \xi_2) \Rightarrow \xi_2)) \rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$;

and $R_\ell = R_g$. ∇

Next we present the notion of modulated Hilbert calculus morphism. In the sequel we consider modulated Hilbert calculi $H = \langle \Sigma, R_g, R_\ell \rangle$, possibly with superscripts.

Definition 8.3.6 A *modulated Hilbert calculus morphism* $h : H \to H'$ is a pair $h = \langle \hat{h}, \check{h} \rangle$ where

- $\hat{h} : \Sigma \to \Sigma'$ is a modulated signature morphism such that
  - $\hat{h}(r) \in {R_g}'$ for every $r \in R_g$;
  - $\hat{h}(r) \in {R_\ell}'$ for every $r \in R_\ell$;
- $\check{h} : gL(\Sigma') \to gL(\Sigma)$ is a map such that
  - $\vdash^g_H \check{h}(\psi')$ whenever $\vdash^g_{H'} \psi'$ and $\check{h}(\varphi') \vdash^g_H \check{h}(\psi')$ whenever $\varphi' \vdash^g_{H'} \psi'$;
  - $\check{h}(\varphi') \vdash^\ell_H \check{h}(\psi')$ whenever $\varphi' \vdash^\ell_{H'} \psi'$;
  - $\check{h}$ is left adjoint of $\hat{h}$;

– $(\check{h}(\varphi'_1)\&\check{h}(\varphi'_2)) = \check{h}(\varphi'_1\&'\varphi'_2)$;
– $\hat{h}(c(\check{h}(\varphi'_1),\ldots,\check{h}(\varphi'_k))) \vdash^{\ell}_{H'} \hat{h}(c)(\varphi'_1,\ldots,\varphi'_k)$,
where c is any constructor in C. ∇

A modulated Hilbert calculus morphism $h : H \to H'$ is a pair. The first component, $\hat{h}$, is a modulated signature morphism. The second component, $\check{h}$, is a (contravariant) map that relates ground formulas of both calculi. For a diagrammatic perspective see Figure 8.14.

$$H \xrightarrow{\ h\ } H'$$

$$\Sigma \xrightarrow{\ \hat{h}\ } \Sigma' \qquad\qquad gL(\Sigma) \xleftarrow{\ \check{h}\ } gL(\Sigma')$$

Figure 8.14: Components of Hilbert calculus morphism

These two components impose several conditions on both calculi. The translation of every global inference rule in H is also a global inference rule in H'. Similarly with respect to local inference rules in H. As in the case of higher-order Hilbert calculi introduced in Chapter 7, we could require a weaker condition, namely, that the translation of the conclusion of each rule is derivable from the translation of the corresponding set of premises. As mentioned therein, noting changes with respect to fibring when we adopt this stronger notion of morphism. The maps $\hat{h}'$ and $\hat{h}''$ preserve theorems and global and local derivations from singletons. The left adjoint condition on $\check{h}$ means that

$$\varphi' \vdash^{\ell}_{H'} \hat{h}(\check{h}(\varphi')) \text{ and } \check{h}(\hat{h}(\varphi)) \vdash^{\ell}_{H} \varphi$$

for every $\varphi' \in gL(\Sigma')$ and $\varphi \in gL(\Sigma)$. This map can be seen as a map relating truth values (formulas) in the Lindendaum-Tarski algebras that will be discussed when studying preservation properties.

The more complex notion of Hilbert calculus morphism turns out to be the adequate one for fulfilling the requirements that are necessary for preserving congruence by fibring later on. Recall that preservation of congruence was important for establishing completeness preservation, in Chapter 3. Therein, preservation of congruence was obtained by sharing implication and equivalence. This may not be now the best solution, since sharing of implication and equivalence leads in most cases to collapse.

The following facts will be useful later on.

Lemma 8.3.7 *Let $h : H \to H'$ be a modulated Hilbert calculus morphism.*

1. *If $\check{h}$ is surjective then $\check{h}(\hat{h}(\varphi)) \cong_H \varphi$, for every $\varphi \in gL(\Sigma)$.*
2. *$\check{h}(\Gamma') \vdash^{g}_{H} \check{h}(\varphi')$ whenever $\Gamma' \vdash^{g}_{H'} \varphi'$, for every $\Gamma' \cup \{\varphi'\} \subseteq gL(\Sigma')$.*

3. $\check{h}(\Gamma') \vdash^{\ell}_{H} \check{h}(\varphi')$ *whenever* $\Gamma' \vdash^{\ell}_{H'} \varphi'$, *for every* $\Gamma' \cup \{\varphi'\} \subseteq gL(\Sigma')$.

Proof.

1. The left adjoint condition ensures that $\check{h}(\hat{h}(\varphi)) \vdash^{\ell}_{H} \varphi$. Since $\check{h}$ is surjective, there is $\varphi' \in gL(\Sigma')$ such that $\check{h}(\varphi') = \varphi$. Using the left adjoint condition,

$$\varphi' \vdash^{\ell}_{H'} \hat{h}(\check{h}(\varphi')).$$

By the property of $\check{h}$ (see Definition 8.3.6), it follows that $\check{h}(\varphi') \vdash^{\ell}_{H} \check{(}\hat{h}(\check{h}(\varphi')))$, that is, $\varphi \vdash^{\ell}_{H} \check{h}(\hat{h}(\varphi))$.

2. Using compactness, if $\Gamma' \vdash^{g}_{H'} \varphi'$, then there is a finite set $\Psi' \subseteq \Gamma'$ such that $\Psi' \vdash^{g}_{H'} \varphi'$. If the result holds for finite sets, then $\check{h}(\Psi') \vdash^{g}_{H} \check{h}(\varphi')$ and, using, monotonicity,

$$\check{h}(\Gamma') \vdash^{g}_{H} \check{h}(\varphi').$$

We now prove that the result holds for finite sets. When Γ' is empty or singular, the result follows from Definition 8.3.6. Assume now that $\Gamma' = \{\varphi'_1, \ldots, \varphi'_k\}$, with $k \geq 2$.

In the following recall that each local derivation is also a global derivation and observe that, in every modulated Hilbert calculus H, $\{(\psi_1 \& \ldots \& \psi_k)\} \vdash^{\ell}_{H} \psi_j$ and $\{\psi_1, \ldots, \psi_k\} \vdash^{\ell}_{H} (\psi_1 \& \ldots \& \psi_k)$, $k \geq 2$. Hence, $(\varphi'_1 \&' \ldots \&' \varphi'_k) \vdash^{\ell}_{H'} \varphi'_j$, for each $1 \leq j \leq k$. Since, $\{\varphi'_1, \ldots, \varphi'_k\} \vdash^{\ell}_{H'} \varphi'$, we then conclude

$$(\varphi'_1 \&' \ldots \&' \varphi'_k) \vdash^{\ell}_{H'} \varphi'.$$

By Definition 8.3.6, $\check{h}(\varphi'_1 \&' \ldots \&' \varphi'_k) \vdash^{\ell}_{H} \check{h}(\varphi')$. Moreover,

$$\{\check{h}(\varphi'_1), \ldots, \check{h}(\varphi'_k)\} \vdash^{\ell}_{H} (\check{h}(\varphi'_1) \& \ldots \& \check{h}(\varphi'_k)).$$

Since $(\check{h}(\varphi'_1) \& \ldots \& \check{h}(\varphi'_k)) = \check{h}(\varphi'_1 \&' \ldots \&' \varphi'_k)$, we conclude that

$$\{\check{h}(\varphi'_1), \ldots, \check{h}(\varphi'_k)\} \vdash^{\ell}_{H} \check{h}(\varphi').$$

3. The proof is similar to the one presented above. ◁

We now prove that derivations are preserved by modulated Hilbert calculus morphisms, when they are injective for schema variables and verify a condition involving safe-relevant morphisms.

Proposition 8.3.8 *Let* $h : H \to H'$ *be a modulated Hilbert calculus morphism such that* $\hat{h}$ *is injective for* Ξ *and* $\hat{h}(C) \subseteq s'(\breve{C}')$ *whenever* $\hat{h}(\Xi) \cap s'(\breve{\Xi}') \neq \emptyset$ *for every* $s' : \breve{\Sigma}' \to \Sigma'$ *in* S'. *Let* $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$ *and further assume that* $id_{\Sigma} \in S$ *if some formula in* $\Gamma \cup \{\varphi\}$ *involves schema variables.*

1. *If* $\Gamma \vdash^{g}_{H} \varphi$ *then* $\hat{h}(\Gamma) \vdash^{g}_{H'} \hat{h}(\varphi)$.

2. *If* $\Gamma \vdash^{\ell}_{H} \varphi$ *then* $\hat{h}(\Gamma) \vdash^{\ell}_{H'} \hat{h}(\varphi)$.

Proof. It is easily established by induction that if σ is a substitution over Σ then the substitution $\hat{h}(\sigma)$ over Σ', defined as $\hat{h}(\sigma)(\hat{h}(\xi)) = \hat{h}(\sigma(\xi))$ and $\hat{h}(\sigma)(\xi') = \xi'$ whenever $\xi' \in \Xi' \setminus \hat{h}(\Xi)$, is such that

$$\hat{h}(\sigma(\psi)) = \hat{h}(\sigma)(\hat{h}(\psi))$$

for every $\psi \in L(\Sigma)$. Note that $\hat{h}(\sigma)$ is well defined because $\hat{h}$ is injective for Ξ.

We now prove that if $\mathrm{id}_\Sigma \in S$ and σ is a substitution over Σ safe for Φ then the substitution $\hat{h}(\sigma)$ over Σ', defined as above, is safe for $\hat{h}(\Phi)$. Consider $s' : \breve{\Sigma}' \to \Sigma'$ in S' and $\breve{\xi}' \in \breve{\Xi}'$ such that

$$s'(\breve{\xi}') \in \mathrm{Var}(\hat{h}(\Phi)) = \hat{h}(\mathrm{Var}(\Phi)) \subseteq \hat{h}(\Xi).$$

Let us consider

$$\hat{h}(\sigma)(s'(\breve{\xi}')) = \hat{h}(\sigma)(\hat{h}(\xi)) = \hat{h}(\sigma(\xi))$$

where $\xi \in \Xi$ is such that $s'(\breve{\xi}') = \hat{h}(\xi)$. Since $\hat{h}(\xi) = s'(\breve{\xi}') \in \hat{h}(\mathrm{Var}(\Phi))$ and $\hat{h}$ is injective, $\xi \in \mathrm{Var}(\Phi)$. Moreover, since σ is safe for Φ, $\mathrm{id}_\Sigma \in S$ and $\mathrm{id}_\sigma(\xi) = \xi \in \mathrm{Var}(\Phi)$ then

$$\sigma(\mathrm{id}_\Sigma(\xi)) = \sigma(\xi) \in L(\Sigma, \mathrm{id}_\Sigma)$$

that is,

$$\sigma(\xi) = c(\varphi_1, \ldots, \varphi_k)$$

for some constructor in Σ (including &). From $s'(\breve{\xi}') \in \hat{h}(\Xi)$ it follows that $\hat{h}(C) \subseteq s'(\breve{C}')$. Thus, $\hat{h}(\sigma(\xi)) = \hat{h}(c)(\hat{h}(\varphi_1), \ldots, \hat{h}(\varphi_k)) \in L(\Sigma', s')$. Therefore $\hat{h}(\sigma)$ is safe for $\hat{h}(\Phi)$.

Observe that if σ_1, σ_2 are substitutions over Σ and σ_1 is safe for Γ, then the substitution $\sigma_2 \circ \sigma_1$ such that

$$\sigma_2 \circ \sigma_1(\xi) = \sigma_2(\sigma_1(\xi))$$

for each $\xi \in \Xi$ is also safe for Γ. Using this result it is easy to prove that, for every substitution σ over Σ, if $\varphi_1 \ldots \varphi_n$ is a global (local) derivation from Γ in H then $\sigma(\varphi_1) \ldots \sigma(\varphi_n)$ is also a global (local) derivation from Γ in H. Hence, if $\Gamma \vdash^g_H \varphi$ and $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$, there is a global derivation of φ from Γ where all the formulas are in $gL(\Sigma)$. Similarly with respect to local derivations.

In the following we prove 1. and 2. by induction on the length of a derivation of φ from Γ.

1. Base: Let $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$ and assume there is a derivation of φ from Γ with length 1. If $\varphi \in \Gamma$, $\hat{h}(\varphi) \in \hat{h}(\Gamma)$ and we are done. Otherwise, there is $\langle \emptyset, \delta \rangle \in R_g$ and a safe substitution σ for δ such that, $\sigma(\delta) = \varphi$. From Definition 8.3.6, $\langle \emptyset, \hat{h}(\delta) \rangle \in {R_g}'$. Consider the substitution $\hat{h}(\sigma)$ as defined above. We have that

$$\hat{h}(\sigma)(\hat{h}(\delta)) = \hat{h}(\sigma(\delta)) = \hat{h}(\varphi).$$

If φ involves a schema variable, $\mathrm{id}_\Sigma \in S$ thus, as proved above, $\hat{h}(\sigma)$ is safe for $\hat{h}(\delta)$.

We now prove that $\hat{h}(\sigma)$ is safe for $\hat{h}(\delta)$ when $\varphi \in gL(\Sigma)$.

(i) Given $s' \in S'$ and $s'(\breve{\xi}) \in \mathrm{Var}(\hat{h}(\delta)) = \hat{h}(\mathrm{Var}(\delta)) \subseteq \hat{h}(\Xi)$, we have that

$$\hat{h}(\sigma)(s'(\breve{\xi})) = \hat{h}(\sigma)(\hat{h}(\xi)) = \hat{h}(\sigma(\xi))$$

for some $\xi \in \mathrm{Var}(\delta)$ and $\hat{h}(C) \subseteq s'(\breve{C}')$.

(ii) We can assume that $\sigma(\xi) \notin \Xi$, hence $\sigma(\xi) = c(\varphi_1, \ldots, \varphi_k)$ for some constructor in Σ and therefore

$$\hat{h}(\sigma(\xi)) = \hat{h}(c)(\hat{h}(\varphi_1), \ldots, \hat{h}(\varphi_k)) \in L(\Sigma', s').$$

Hence, $\hat{h}(\Gamma) \vdash^g_{H'} \hat{h}(\varphi)$.

Step: Let $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$ and assume there is a derivation $\varphi_1 \ldots \varphi_n$ of φ from Γ with $n > 1$ (recall that φ_n is φ). In the interesting case, there is $\langle \Delta, \delta \rangle \in R_g$ and a safe substitution σ for $\Delta \cup \{\delta\}$ such that, $\sigma(\delta) = \varphi$ and $\sigma(\Delta) \subseteq \{\varphi_1, \ldots, \varphi_{n-1}\}$. Consider $\hat{h}(\sigma)$ as above.

If some formula in $\Gamma \cup \{\varphi\}$ involves a schema variable, $\mathrm{id}_\Sigma \in S$ thus, again, $\hat{h}(\sigma)$ is safe for $\hat{h}(\Delta \cup \{\delta\})$.

Assume now that $\Gamma \cup \{\varphi\} \subseteq gL(\Sigma)$. Given $s' \in S'$, and

$$s'(\breve{\xi}) \in \mathrm{Var}(\hat{h}(\Delta \cup \{\delta\})) = \hat{h}(\mathrm{Var}(\Delta \cup \{\delta\})) \subseteq \hat{h}(\Xi)$$

we have

$$\hat{h}(\sigma)(s'(\breve{\xi})) = \hat{h}(\sigma)(\hat{h}(\xi)) = \hat{h}(\sigma(\xi))$$

for some $\xi \in \mathrm{Var}(\Delta \cup \{\delta\})$ and $\hat{h}(C) \subseteq s'(\breve{C}')$. As remarked above, we can assume that $\varphi_1, \ldots, \varphi_n \in gL(\Sigma)$ and it is again possible to choose σ such that $\sigma(\xi) \notin \Xi$. Thus,

$$\hat{h}(\sigma(\xi)) = \hat{h}(c)(\hat{h}(\varphi_1), \ldots, \hat{h}(\varphi_k)).$$

Hence, $\hat{h}(\sigma(\xi)) \in L(\Sigma', s')$ and therefore $\hat{h}(\sigma)$ is safe for $\hat{h}(\Delta \cup \{\delta\})$.

From Definition 8.3.6, $\langle \hat{h}(\Delta), \hat{h}(\delta) \rangle \in {R_g}'$. We have that $\hat{h}(\sigma)(\hat{h}(\Delta)) = \hat{h}(\sigma(\Delta))$, $\hat{h}(\sigma(\Delta)) \subseteq \{\hat{h}(\varphi_1), \ldots, \hat{h}(\varphi_{n-1})\}$ and $\hat{h}(\sigma)(\hat{h}(\delta)) = \hat{h}(\sigma(\delta)) = \hat{h}(\varphi)$. By the induction hypothesis

$$\hat{h}(\Gamma) \vdash^g_{H'} \hat{h}(\varphi_i)$$

for $1 \leq i \leq n-1$. Therefore, $\hat{h}(\Gamma) \vdash^g_{H'} \hat{h}(\varphi)$.

2. The proof uses 1. and is similar to the proof above. ◁

As expected, modulated Hilbert calculi and their morphisms constitute a category.

Prop/Definition 8.3.9 *Modulated Hilbert calculi and their morphisms constitute the category* ***mHil****.*

Proof. Straightforward, using Definitions 8.3.3 and 8.3.6 and Proposition 8.3.8. $\triangleleft$

We can now describe the modulated fibring mechanism for (modulated) Hilbert calculi. As previously done for interpretation systems, we must start by defining a bridge. The bridge allows a mild relationship between the formulas in the modulated Hilbert calculi that we want to combine as well as between their consequence relations. Again modulated fibring appears as a pushout in the category of modulated Hilbert calculi and their morphisms.

Definition 8.3.10 A *bridge* between modulated Hilbert calculi H' and H'' is a diagram

$$\beta = \langle f' : \breve{H} \to H', f'' : \breve{H} \to H'' \rangle$$

in **mHil** such that

- $\breve{\Xi} = \emptyset$;
- $S' = S'' = \emptyset$ or $\mathrm{id}_{\Sigma'} \in S'$ and $\mathrm{id}_{\Sigma''} \in S''$;
- $\hat{f}'$ and $\hat{f}''$ are injective and $\check{f}'$ and $\check{f}''$ are surjective. ∇

A bridge between modulated Hilbert calculi H' and H'' (see Figure 8.15) consists of two modulated Hilbert calculus morphisms f' and f'' from a calculus $\breve{H}$ respectively to H' and H''.

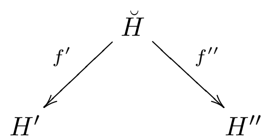


Figure 8.15: Bridge of Hilbert calculi

The covariant components of f' and f'' are injective and the contravariant components of f' and f'' are surjective. The conditions on $\breve{H}$, H' and H'' ensure that the modulated fibring can be obtained as a pushout of the bridge.

Prop/Definition 8.3.11 *The* modulated fibring *of modulated Hilbert calculi H' and H'' by a bridge β is a pushout of β in* **mHil**, *if the pushout of β exists.*

Proof. Consider the bridge $\beta = \langle f' : \breve{H} \to H', f'' : \breve{H} \to H'' \rangle$. The pushout

$$\langle g' : H' \to H, g'' : H'' \to H \rangle$$

of β is defined as follows (see Figure 8.16):

- $\langle \hat{g}' : \Sigma' \to \Sigma, \hat{g}'' : \Sigma'' \to \Sigma \rangle$ is a pushout in **Sig** of $Sg(\beta)$;

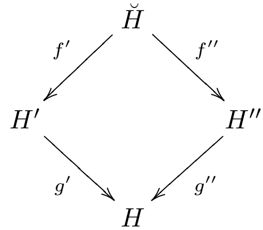


Figure 8.16: Pushout of a bridge of Hilbert calculi

- $\check{g}'$ and $\check{g}''$ are inductively defined as follows where, for simplicity, we only include the inductive steps related to $\check{g}'$ (the ones for $\check{g}''$ are similar):
  - $\check{g}'(\hat{g}'(c')) = c'$ and $\check{g}'(\hat{g}''(c'')) = \hat{f}'(\check{f}''(c''))$;
  - $\check{g}'(\hat{g}'(c')(\vec{\varphi})) = c'(\check{g}'(\vec{\varphi}))$ and $\check{g}'(\hat{g}''(c'')(\vec{\varphi})) = \hat{f}'(\check{f}''(c''(\check{g}''(\vec{\varphi}))))$;
  - $\check{g}'(\varphi_1 \& \varphi_2) = \check{g}'(\varphi_1) \&' \check{g}'(\varphi_2)$;
- R_ℓ includes $\hat{g}'(R_\ell) \cup \hat{g}''(R_\ell{}'')$, & elimination and introduction plus the following rules, for any $\delta \in gL(\Sigma)$ and $\vec{\delta}$ a sequence over $gL(\Sigma)$:
  - $\langle \{\delta\}, \hat{g}'(\check{g}'(\delta)) \rangle$;
  - $\langle \{\hat{g}'(c')(\hat{g}'(\check{g}'(\vec{\delta})))\}, \hat{g}'(c')(\vec{\delta}) \rangle$;
  - similar rules for $\hat{g}''$ and $\check{g}''$;
- $R_g = \hat{g}'(R_g{}') \cup \hat{g}''(R_g{}'') \cup R_\ell$.

To prove that $\langle g' : H' \to H, g'' : H'' \to H \rangle$ is indeed a pushout in **mHil** of f' and f'', it is important to take into account the following facts. Since $\breve{\Xi} = \emptyset$, we can prove that $\hat{f}'$, $\hat{f}''$, $\hat{g}'$ and $\hat{g}''$ are monotonic. Observe also that

$$\check{g}'(\hat{g}'(\varphi')) = \varphi'$$

for each $\varphi' \in gL(\Sigma')$. The same applies to $\check{g}''$. Moreover,

$$\check{g}''(\hat{g}'(\varphi')) = \hat{f}''(\check{f}'(\varphi'))$$

for each $\varphi' \in gL(\Sigma')$ is easily established by induction. Similarly,

$$\check{g}'(\hat{g}''(\varphi'')) = \hat{f}'(\check{f}''(\varphi''))$$

for each $\varphi'' \in gL(\Sigma'')$.

1. We first prove that $g' \circ f' = g'' \circ f''$. From the pushout in **mSig** we get $\hat{g}' \circ \hat{f}' = \hat{g}'' \circ \hat{f}''$. We prove

$$\check{f}''(\check{g}''(\varphi)) \cong_{\check{H}} \check{f}'(\check{g}'(\varphi))$$

for each $\varphi \in gL(\Sigma)$, by structural induction. Recall that $\check{f}'$ and $\check{f}''$ are surjective.

Base: If $\varphi = \hat{g}'(c') = \hat{g}''(c'')$, there is $\check{c} \in \check{C}_0$ such that

$$\hat{f}'(\check{c}) = c' \text{ and } \hat{f}''(\check{c}) = c''.$$

Hence, $\check{f}'(c') = \check{f}'(\hat{f}'(\check{c}))$ and $\check{f}''(c'') = \check{f}''(\hat{f}''(\check{c}))$. Using Lemma 8.3.7, $\check{f}'(c') \cong_{\check{H}} \check{c}$ and $\check{f}''(c'') \cong_{\check{H}} \check{c}$. Since $\check{f}''(\check{g}''(\varphi)) = \check{f}''(c'')$ and $\check{f}'(\check{g}'(\varphi)) = \check{f}'(c')$, it follows that

$$\check{f}''(\check{g}''(\varphi)) \cong_{\check{H}} \check{f}'(\check{g}'(\varphi)).$$

If $\varphi = \hat{g}'(c')$ and $\varphi \notin \hat{g}''(C''_0)$, then we have $\check{f}''(\check{g}''(\varphi)) = \check{f}''(\hat{f}''(\check{f}'(c')))$ and $\check{f}'(c') = \check{f}'(\check{g}'(\varphi))$. Using again Lemma 8.3.7,

$$\check{f}''(\check{g}''(\varphi)) \cong_{\check{H}} \check{f}'(\check{g}'(\varphi)).$$

The case $\varphi = \hat{g}''(c'') \notin \hat{g}'(C'_0)$ is similar.

Step: If $\varphi = \hat{g}'(c')(\vec{\varphi})$ where $\hat{g}'(c') = \hat{g}''(c'')$, then

$$\check{f}'(\check{g}'(\hat{g}'(c')(\vec{\varphi}))) = \check{f}'(c'(\check{g}'(\vec{\varphi}))) \text{ and } \check{f}''(\hat{f}''(\check{f}'(c'(\check{g}'(\vec{\varphi}))))) = \check{f}''(\check{g}''(\hat{g}'(c')(\vec{\varphi}))).$$

Hence, again by Lemma 8.3.7,

$$\check{f}'(\check{g}'(\hat{g}'(c')(\vec{\varphi}))) \cong_{\check{H}} \check{f}''(\check{g}''(\hat{g}'(c')(\vec{\varphi}))).$$

If $\varphi = \hat{g}'(c')(\vec{\varphi})$ where $\hat{g}'(c') \notin \hat{g}''(C''_k)$, following a similar reasoning we can conclude that

$$\check{f}''(\check{g}''(\hat{g}'(c')(\vec{\varphi})) \cong_{\check{H}} \check{f}'(\check{g}'(\hat{g}'(c')(\vec{\varphi}))).$$

The case $\varphi = \hat{g}''(c'') \notin \hat{g}'(C'_k)$ is again similar.

Finally, the case $\varphi = (\varphi_1 \& \varphi_2)$ easily follows from the induction hypothesis, and the definitions of modulated Hilbert calculi morphism, $\check{g}'$ and $\check{g}''$.

2. We now refer to the proof that g' and g'' and indeed morphisms in **mHil**. Most conditions follow straightforwardly from the definition of $\check{g}'$ and $\check{g}''$.

The preservation of derivations by $\check{g}'$ and $\check{g}''$ follows by induction in the length of derivations and uses the fact that given a substitution $\sigma : \Xi \to L(\Sigma)$ safe for $\hat{g}'(\Gamma')$ then the substitution $\sigma' : \Xi' \to L(\Sigma')$ such that

$$\sigma'(\xi') = \check{g}'(\sigma(\hat{g}'(\xi')))$$

is safe for Γ' and

$$\sigma'(\varphi') = \check{g}'(\sigma(\hat{g}'(\varphi'))).$$

A similar substitution $\sigma'' : \Xi'' \to L(\Sigma'')$ is also considered.

The proof also uses the requirements on S' and S'', the properties of the morphisms f' and f'' and, at some point of the step, it also uses 1.

3. Universal property. Let $h' : H' \to H'''$ and $h'' : H'' \to H'''$ be Hilbert calculus morphisms such that $h' \circ f' = h'' \circ f''$.

Existence. Let $h : H \to H'''$ be as follows:

- $\hat{h}$ is the unique morphism in **Sig** such that $\hat{h} \circ \hat{g}' = \hat{h}'$;
- $\hat{h} \circ \hat{g}'' = \hat{h}''$;
- $\check{h}$ is such that $\check{h}(\varphi''') = \hat{g}'(\check{h}'(\varphi'''))\&\hat{g}''(\check{h}''(\varphi'''))$.

It is straightforward to show that h is a Hilbert calculus morphism. The proof uses, in particular, monotonicity of $\hat{g}'$ and $\hat{g}''$.

Next we prove that $\check{g}'(\check{h}(\varphi''')) \cong^{\ell}_{H'} \check{h}'(\varphi''')$.

(i) We have that

$$\begin{aligned}\check{g}'(\check{h}(\varphi''')) &= \check{g}'(\hat{g}'(\check{h}'(\varphi'''))\&\hat{g}''(\check{h}''(\varphi''')))\\ &= \check{g}'(\hat{g}'(\check{h}'(\varphi''')))\&'\check{g}'(\hat{g}''(\check{h}''(\varphi''')))\end{aligned}$$

thus, using $\&'$ elimination, it follows that

$$\check{g}'(\check{h}(\varphi''')) \vdash^{\ell}_{H'} \check{g}'(\hat{g}'(\check{h}'(\varphi''')))$$

that is, $\check{g}'(\check{h}(\varphi''')) \vdash^{\ell}_{H'} \check{h}'(\varphi''')$.

(ii) We have that $\check{h}'(\varphi''') \vdash^{\ell}_{H'} \hat{f}'(\check{f}'(\check{h}'(\varphi''')))$. Since

$$\check{f}'(\check{h}'(\varphi''')) \cong^{\ell}_{\check{H}} \check{f}''(\check{h}''(\varphi'''))$$

using Lemma 8.3.8, we get

$$\hat{f}'(\check{f}'(\check{h}'(\varphi'''))) \cong^{\ell}_{H'} \hat{f}'(\check{f}''(\check{h}''(\varphi''')))$$

that is,

$$\hat{f}'(\check{f}'(\check{h}'(\varphi'''))) \cong^{\ell}_{H'} \check{g}'(\hat{g}''(\check{h}''(\varphi'''))).$$

Therefore, $\check{h}'(\varphi''') \vdash^{\ell}_{H'} \check{g}'(\hat{g}''(\check{h}''(\varphi''')))$. Since, $\check{h}'(\varphi''') = \check{g}'(\hat{g}'(\check{h}'(\varphi''')))$, using $\&'$ introduction,

$$\check{h}'(\varphi''') \vdash^{\ell}_{H'} \check{g}'(\hat{g}'(\check{h}'(\varphi''')))\&'\check{g}'(\hat{g}''(\check{h}''(\varphi''')))$$

that is, $\check{h}'(\varphi''') \vdash^{\ell}_{H'} \check{g}'(\check{h}(\varphi'''))$.

Uniqueness. Let $k : H \to H'''$ be such that $k \circ g' = h'$ and $k \circ g'' = h''$. The equality $\hat{h}(\varphi) = \hat{k}(\varphi)$, for each $\varphi \in L(\Sigma)$, easily follows by induction.

We now prove that

$$\check{k}(\varphi''') \cong^{\ell}_{H} \check{h}(\varphi''')$$

for each $\varphi \in gL(\Sigma)$.

We first prove that

$$\check{k}(\varphi''') \cong^{\ell}_{H} \hat{g}'(\check{g}'(k(\varphi''')))\&\hat{g}''(\check{g}''(k(\varphi'''))).$$

(i) We have that $\check{k}(\varphi''') \vdash^{\ell}_{H} \hat{g}'(\check{g}'(k(\varphi''')))$ and $\check{k}(\varphi''') \vdash^{\ell}_{H} \hat{g}''(\check{g}''(k(\varphi''')))$. Thus,

$$\check{k}(\varphi''') \vdash^{\ell}_{H} \hat{g}'(\check{g}'(k(\varphi''')))\&\hat{g}''(\check{g}''(k(\varphi'''))).$$

(ii) Assuming that $k(\varphi''') = \hat{g}'(c')(\vec{\varphi})$, we have

$$\hat{g}'(\check{g}'(\hat{g}'(c')(\vec{\varphi})))\&\hat{g}''(\check{g}''(\hat{g}'(c')(\vec{\varphi}))) \vdash^{\ell}_{H} \hat{g}'(\check{g}'(\hat{g}'(c')(\vec{\varphi}))).$$

Since $\hat{g}'(\check{g}'(\hat{g}'(c')(\vec{\varphi}))) = \hat{g}'(c'(\check{g}'(\vec{\varphi})))$ and $\hat{g}'(c'(\check{g}'(\vec{\varphi}))) \vdash^{\ell}_{H} \hat{g}'(c')(\vec{\varphi})$ we conclude

$$\hat{g}'(\check{g}'(k(\varphi''')))\&\hat{g}''(\check{g}''(k(\varphi'''))) \vdash^{\ell}_{H} \check{k}(\varphi''')$$

in this case. A similar conclusion also holds when we consider $k(\varphi''') = \hat{g}''(c'')(\vec{\varphi})$. Hence

$$\hat{g}'(\check{g}'(k(\varphi''')))\&\hat{g}''(\check{g}''(k(\varphi'''))) \vdash^{\ell}_{H} \check{k}(\varphi''').$$

Using & elimination,

$$\check{h}(\varphi''') \vdash^{\ell}_{H} \hat{g}'(\check{h}'(\varphi''')) \text{ and } \check{h}(\varphi''') \vdash^{\ell}_{H} \hat{g}''(\check{h}''(\varphi''')).$$

Moreover, $k \circ g' = h'$ and $k \circ g'' = h''$, thus

$$\hat{g}'(\check{g}'(k(\varphi'''))) \cong^{\ell}_{H} \hat{g}'(\check{h}'(\varphi'''))$$

and

$$\hat{g}''(\check{g}''(k(\varphi'''))) \cong^{\ell}_{H} \hat{g}''(\check{h}''(\varphi''')).$$

Therefore, using & introduction, it follows that

$$\check{h}(\varphi''') \vdash^{\ell}_{H} \hat{g}'(\check{g}'(k(\varphi''')))\&\hat{g}''(\check{g}''(k(\varphi''')))$$

and we conclude that

$$\check{h}(\varphi''') \vdash^{\ell}_{H} \check{k}(\varphi''').$$

Reasoning in a similar way, using the fact that $\check{k}(\varphi''') \vdash^{\ell}_{H} \hat{g}'(\check{g}'(k(\varphi''')))$ and $\check{k}(\varphi''') \vdash^{\ell}_{H} \hat{g}''(\check{g}''(k(\varphi''')))$ we conclude $\check{k}(\varphi''') \vdash^{\ell}_{H} \check{h}(\varphi''')$. Hence, $\check{k}(\varphi''') \cong^{\ell}_{H} \check{h}(\varphi''')$. ◁

Observe that it may be the case that there is no pushout of a given bridge β. We illustrate this situation in Example 8.3.12.

Example 8.3.12 Consider the bridge $\beta = \langle f' : \breve{H} \to H', f'' : \breve{H} \to H''\rangle$, presented in Figure 8.17, where

- $C'_0 = \{d', e'\}$, $C'_1 = \{c'\}$ and $C'_k = \emptyset$ for $k \geq 2$;
- $C''_0 = \{d'', e''\}$, $C''_1 = \{c''\}$ and $C''_k = \emptyset$ for $k \geq 2$, $\breve{C}_k = \emptyset$ for $k \geq 2$, and $\breve{\Xi} = \emptyset$;
- $\hat{f}'(d) = d'$, $\hat{f}'(c) = c'$ and $\hat{f}''(d) = d''$, $\hat{f}''(c) = c''$;

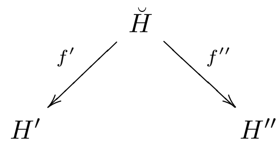


Figure 8.17: Example of a bridge of Hilbert calculi

- $\breve{f}'(d') = \breve{f}''(d'') = f'(e') = \breve{f}''(e'') = d$,
  $\breve{f}'(c'(\varphi')) = (c(\breve{f}'(\varphi'))$ and $\breve{f}''(c''(\varphi'')) = (c(\breve{f}''(\varphi''))$.

Let us follow Prop/Definition 8.3.11 to get a pushout

$$\langle g' : H' \to H, g'' : H'' \to H\rangle$$

of β (see Figures 8.18 and 8.19. First,

$$\langle \hat{g}' : \Sigma' \to \Sigma, \hat{g}'' : \Sigma'' \to \Sigma\rangle$$

is a pushout of $\langle \hat{f}' : \breve{\Sigma} \to \Sigma', \hat{f}'' : \breve{\Sigma} \to \Sigma''\rangle$ in **mSig**. Hence, we can consider

- $C_0 = \{d'|d'', e', e''\}$, $C_1 = \{c'|c''\}$;
- $\hat{g}'(c') = \hat{g}''(c'') = c'|c''$, $\hat{g}'(d') = \hat{g}''(d'') = d'|d''$, $\hat{g}'(e') = e'$ and $\hat{g}''(e'') = e''$.

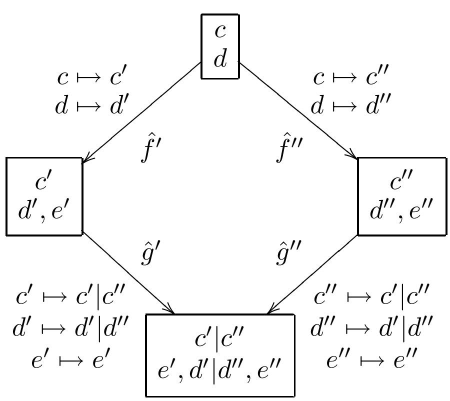


Figure 8.18: Example of a pushout of Hilbert calculi

We now turn our attention to the maps

$$\breve{g}' : gL(\Sigma) \to gL(\Sigma') \text{ and } \breve{g}'' : gL(\Sigma) \to gL(\Sigma'').$$

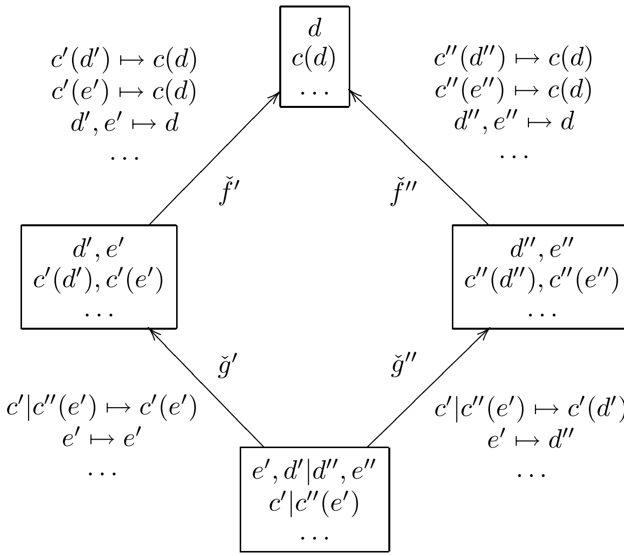


Figure 8.19: Example of a pushout of Hilbert calculi (continued)

In particular, $\check{g}'(e') = \check{g}'(\hat{g}'(e')) = e'$ and $\check{g}''(e') = \check{g}''(\hat{g}'(e')) = \hat{f}''(\check{f}'(e')) = \hat{f}''(d) = d''$.

Lets consider now $\check{g}'(c'|c''(e'))$. From Prop/Definition 8.3.11, two conditions on $\check{g}'(c'|c''(e'))$ should hold. On one hand,

$$\begin{aligned}\check{g}'(c'|c''(e')) &= \check{g}'(\hat{g}'(c')(e'))\\ &= \hat{f}'(\check{f}''(c''(\hat{g}''(e')\\ &= \hat{c}'(\hat{g}'(e'))\\ &= c'(e')\end{aligned}$$

On the other hand,

$$\begin{aligned}\check{g}'(c'|c''(e')) &= \check{g}'(\hat{g}''(c'')(e'))\\ &= \hat{f}'(\check{f}''(c''(\hat{g}''(e')\\ &= \hat{f}'(\check{f}''(c''(d''))))\\ &= \hat{f}'(c(d)) = c'(d')\end{aligned}$$

However, in this case, the two values are different, and, as a consequence, the pushout of β does not exist. $\triangledown$

Next we give some examples of modulated fibring of many-valued logics illustrating non-collapsing situations. We start by a general result which states how to choose a bridge without collapsing, when there is no sharing of constructors. As we referred above, this is the case in most situations because otherwise collapsing is inevitable.

Proposition 8.3.13 *Let H', H'' be Hilbert calculi such that*

- $\mathbf{t}' \in C_0'$ *and* $\mathbf{t}'' \in C_0''$;
- C' *and* C'' *are isomorphic in* ***Sig***;
- $id_{\Sigma'} \in S'$ *and* $id_{\Sigma''} \in S''$;

and let $\beta = \langle f' : \breve{H} \to H', f'' : \breve{H} \to H'' \rangle$ *be a bridge such that*

- $\breve{C}_0 = \{\breve{\mathbf{t}}\}$ *and* $\breve{C}_k = \emptyset$ *for all* $k \neq 0$;
- $\breve{\Xi} = \emptyset$ *and* $\breve{S} = \emptyset$;
- $\breve{R}_g = \breve{R}_\ell$ *include* $\{\langle \emptyset, \breve{\mathbf{t}} \rangle\}$ *and the rules for* $\breve{\&}$ *elimination and introduction;*
- $\hat{f}'(\breve{\mathbf{t}}) = \mathbf{t}'$ *and* $\hat{f}''(\breve{\mathbf{t}}) = \mathbf{t}''$.

Then the modulated fibring of H' and H'' by β does not collapse.

Observe that $\check{f}'$ and $\check{f}''$ are constant mappings in the proposition above. This proposition shows that for all cases of unconstrained modulated fibring (that is, only *verum* is shared) it is possible to avoid the collapsing problem. Note that in C_0' and C_0'' we can have propositional symbols. We now sketch the relationships between the signatures and the sets of formulas of the modulated Hilbert systems involved in this modulated fibring in Figure 8.20.

Since $\text{id}_{\Sigma'} \in S'$, $\text{id}_{\Sigma''} \in S''$ we guarantee that all proofs and derivations of the component logics will be proofs and derivations in the modulated fibring. Observe

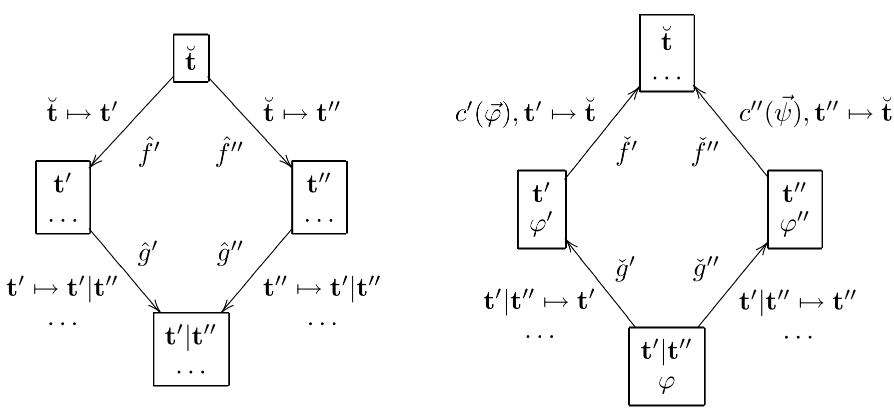


Figure 8.20: Non-collapsing unconstrained fibring

also that the requirement $\mathrm{id}_{\Sigma'} \in S'$, $\mathrm{id}_{\Sigma''} \in S''$ does not change the consequence relations of H' and H''. This requirement only prepares the Hilbert calculi for the combination. We now instantiate Proposition 8.3.13 for several cases.

Example 8.3.14 Assume modulated Hilbert calculi for classical logic, 3-valued Łukasiewicz logic, 3-valued Gödel logic and intuitionistic logic similar to the ones already presented but also including *verum*, the constructors $\wedge$ and $\vee$ of arity 2 and appropriate local derivation rules to cope with *verum* and such that signature identities are safe-relevant morphisms. By choosing the bridge as in Proposition 8.3.13:

(i) we do not get the collapse between classical and 3-valued Łukasiewicz logics;

(ii) we do not get the collapse between classical and 3-valued Gödel logics;

(iii) we avoid the collapsing between classical and intuitionistic logics. ∇

We now show that the example of collapse of classical and intuitionistic logics given in [106] can be avoided in the present context. The example also allows a better understanding of safe substitutions.

Example 8.3.15 Consider the modulated fibring of the modulated Hilbert calculi H' and H'' for intuitionistic logic and classical logic as described in Example 8.3.14, respectively, with the bridge as in Proposition 8.3.13. Then

$$(\hat{g}'(\xi_1')\hat{g}'(\Rightarrow')\hat{g}'(\xi_2')) \vdash_H^g (\hat{g}'(\xi_1')\hat{g}''(\Rightarrow'')\hat{g}'(\xi_2'))$$

does not hold. In particular, the step

$$((\hat{g}'(\xi_1')\hat{g}'(\Rightarrow')\hat{g}'(\xi_2'))\hat{g}''(\wedge'')\hat{g}'(\xi_1')) \vdash_H^g \hat{g}'(\xi_2')$$

is not possible because the underlying substitution for conjunction $\hat{g}''(\wedge'')$ elimination is not safe since $\hat{g}'(\xi_2')$ does not start with a constructor from H''. ∇

We now analyze an example of modulated fibring of Hilbert calculi sharing the negation constructor.

Example 8.3.16 Let H' be the modulated Hilbert calculus for 3-valued Gödel logic and H'' be the Hilbert calculus for classical logic similar to the presented above but also including constructors $\mathbf{t}'$, $\mathbf{t}''$ and $\mathbf{f}'$, $\mathbf{f}''$ of arity 0 and the appropriate derivation rules to cope with this constructors.

Consider a bridge β such that:

- $\breve{C}_0 = \{\breve{\mathbf{f}}, \breve{\mathbf{t}}\}$, $\breve{C}_1 = \{\breve{\neg}\}$, $\breve{C}_k = \emptyset$ for $k \geq 2$, $\breve{\Xi} = \emptyset$, $\breve{S} = \emptyset$;
- $S' = \{\mathrm{id}_{\Sigma'}\}$ and $S'' = \{\mathrm{id}_{\Sigma''}\}$;
- $\hat{f}'(\breve{\mathbf{f}}) = \mathbf{f}'$, $\hat{f}'(\breve{\mathbf{t}}) = \mathbf{t}'$ and $\hat{f}'(\breve{\neg}) = \neg'$;

- $\check{f}'(\varphi') = \begin{cases} \breve{\varphi} & \text{if } \varphi' \text{ is } \hat{f}'(\breve{\varphi}) \\ \breve{\mathbf{f}} & \text{if } \varphi' \vdash^{\ell}_{H'} \mathbf{f}' \\ \breve{\mathbf{t}} & \text{otherwise} \end{cases}$

- $\hat{f}''$ and $\check{f}''$ defined in a similar way;

- $\breve{R}_g$ and $\breve{R}_\ell$ are the translations of the ground instances of $R_g{}'$, $R_g{}''$, $R_\ell{}'$, $R_\ell{}''$ by $\check{f}'$ and $\check{f}''$ plus the rules $\breve{\&}$ elimination and introduction.

We prove that $f' = \langle \hat{f}', \check{f}' \rangle$ is indeed a morphism in **mHil**. The case of f'' is similar.

1. We first prove that $\check{f}'$ is preserves the intended derivations. Let $\varphi' \vdash^{\ell}_{H'} \psi'$. It can be proved that there are derivations for $\varphi' \vdash^{\ell}_{H'} \psi'$ involving only ground formulas. The proof follows by induction on the length of a ground derivation $\varphi'_1 \cdots \varphi'_n$ of ψ' from φ'.

Base. Straightforward.

Step. There is a rule

$$\langle \{\delta'_1, \ldots, \delta'_k\}, \delta' \rangle$$

and a ground substitution σ safe for $\{\delta'_1, \ldots, \delta'_k, \delta'\}$ such that

$$\{\sigma(\delta'_1), \ldots, \sigma(\delta'_k)\} \subseteq \{\varphi'_1, \ldots, \varphi'_{n-1}\}$$

and $\sigma(\delta') = \varphi'_n = \psi'$. By induction hypothesis, $\check{f}'(\varphi') \vdash^{\ell}_{\breve{H}} \check{f}'\sigma(\delta'_i))$ for $i = 1, \ldots, k$. Since

$$\langle \{\check{f}'(\sigma(\delta'_1)), \ldots, \check{f}'(\sigma(\delta'_k))\}, \check{f}'(\sigma(\delta')) \rangle$$

is a rule in $\breve{R}_\ell$, then

$$\check{f}'(\varphi') \vdash^{\ell}_{\breve{H}} \check{f}'(\psi')$$

as intended. The proof is similar in the case of derivations $\vdash^{\ell}_{H'} \psi'$ and in the case of global derivations.

2. We now prove that $\varphi' \vdash^{\ell}_{H'} \hat{f}'(\check{f}'(\varphi'))$.

Assume that φ' is $\hat{f}'(\breve{\varphi})$. Then,

$$\begin{aligned} \hat{f}'(\check{f}'(\varphi')) &= \hat{f}'(\check{f}'(\hat{f}'(\breve{\varphi}))) \\ &= \hat{f}'(\breve{\varphi}) \\ &= \varphi' \end{aligned}$$

and we are done.

Assume now that $\varphi' \vdash^{\ell}_{H'} \mathbf{f}'$. Since $\hat{f}'(\check{f}'(\varphi')) = \hat{f}'(\breve{\mathbf{f}}) = \mathbf{f}'$, we are done.

Otherwise $\hat{f}'(\check{f}'(\varphi')) = \hat{f}'(\breve{\mathbf{t}})) = \mathbf{t}'$. Since $\vdash^{\ell}_{H'} \mathbf{t}'$, we have $\varphi' \vdash^{\ell}_{H'} \hat{f}'(\check{f}'(\varphi'))$.

3. Since $\check{f}'(\hat{f}'(\breve{\varphi})) = \breve{\varphi}$, we conclude that $\check{f}'(\hat{f}'(\breve{\varphi})) \breve{\vdash}^{\ell}{}_H \breve{\varphi}$.

4. Finally, we prove that $\hat{f}'(\breve{\neg}(\check{f}'(\varphi'))) \vdash^{\ell}_{H'} (\neg' \varphi')$.

Assume first that φ' is $\hat{f}'(\breve{\varphi})$. Then we have

$$\hat{f}'(\breve{\neg}(\check{f}'(\varphi'))) = (\overset{\prime}{\neg}\, \varphi').$$

Therefore, $\hat{f}'(\breve{\neg}(\check{f}'(\varphi'))) \vdash^{\ell}_{H'} (\neg' \varphi')$.

Consider now $\varphi' \vdash^{\ell}_{H'} \mathbf{f}'$. In this case,

$$(\neg' \mathbf{f}') \vdash^{\ell}_{H'} (\neg' \varphi') \text{ and } \hat{f}'(\breve{\neg}(\check{f}'(\varphi'))) = \hat{f}'(\breve{\neg}(\breve{\mathbf{f}}) = (\neg' \mathbf{f}').$$

Otherwise, $\check{f}'(\varphi') = \breve{\mathbf{t}}$. Hence $\hat{f}'(\breve{\neg}(\breve{\mathbf{t}})) = (\neg' \hat{f}'(\breve{\mathbf{t}})) = (\neg' \mathbf{t}')$. Since, $(\neg' \mathbf{t}') \vdash^{\ell}_{H'} \mathbf{f}'$ and $\mathbf{f}' \vdash^{\ell}_{H'} (\neg' \varphi')$, we are done.

In the modulated fibring we have $C_k = \hat{g}'(C'_k) \cup \hat{g}''(C''_k)$ and $\Xi = \hat{g}'(\Xi') \cup \hat{g}''(\Xi'')$, $R_g = \hat{g}'({R_g}') \cup \hat{g}''({R_g}'') \cup R_\ell$ and R_ℓ includes $\hat{g}'({R_\ell}') \cup \hat{g}''({R_\ell}'')$, the rules for & elimination and introduction and the rules for the modulated fibring.

Note that we have two forms of detachment:

$$\langle \{\hat{g}'(\xi'_1), (\hat{g}'(\xi'_1)\hat{g}'(\Rightarrow')\hat{g}'(\xi'_2))\}, \hat{g}'(\xi'_2) \rangle$$

and

$$\langle \{\hat{g}''(\xi''_1), (\hat{g}''(\xi''_1)\hat{g}''(\Rightarrow'')\hat{g}''(\xi''_2))\}, \hat{g}''(\xi''_2) \rangle$$

and that we do not have the inference

$$\{\hat{g}'(\varphi'), (\hat{g}'(\varphi')\hat{g}''(\Rightarrow'')\hat{g}''(\varphi''))\} \vdash_p \hat{g}''(\varphi'')$$

for instance, because the substitution of $\hat{g}''(\xi''_1)$ by $\hat{g}'(\varphi')$ is not safe. ▽

We now describe how unconstrained fibring as presented before can be recovered using modulated fibring.

Remark 8.3.17 Let H' and H'' be modulated Hilbert calculi with $S' = S'' = \emptyset$. Consider the following bridge β:

- $\breve{C}_0 = \{\breve{\mathbf{t}}\}$, $\breve{C}_k = \emptyset$ for all $k \neq 0$, $\breve{\Xi} = \emptyset$, $\breve{S} = \emptyset$;
- $\breve{R}_g = \breve{R}_\ell$ include $\{\langle \emptyset, \breve{\mathbf{t}} \rangle\}$ plus & elimination and introduction;
- $\hat{f}'(\breve{\mathbf{t}}) = \mathbf{t}'$ and $\hat{f}''(\breve{\mathbf{t}}) = \mathbf{t}''$.

Observe that the mappings $\check{f}'$ and $\check{f}''$ are constant. Then the modulated fibring of H' and H'' by a bridge β is a conservative extension of the unconstrained fibring as presented in Chapter 2 (with no sharing of constructors except *verum*). ▽

8.4 Modulated logic systems

As done in previous chapters, we can put together the semantic and the deductive components presented above, obtaining modulated logic systems and morphisms between modulated logic systems. As usual, modulated fibring of these logic systems is a pushout in the corresponding category. As before, modulated logic systems constitute the right setting to study the preservation by modulated fibring of properties such as soundness and completeness.

Definition 8.4.1 A *modulated logic system* is a tuple

$$L = \langle \Sigma, M, A, R_g, R_\ell \rangle$$

such that $\langle \Sigma, M, A \rangle$ is a modulated interpretation system and $\langle \Sigma, R_g, R_\ell \rangle$ is a modulated Hilbert calculus. ▽

In the sequel we assume logic systems $L = \langle \Sigma, M, A, R_g, R_\ell \rangle$, possibly with superscripts. $I(L)$ denotes the modulated interpretation system $\langle \Sigma, M, A \rangle$ and $H(L)$ denotes the modulated Hilbert calculus $\langle \Sigma, R_g, R_\ell \rangle$. As usual, we write $\Gamma \vdash_L \varphi$ whenever $\Gamma \vdash_{H(L)} \varphi$ and write $\Gamma \vDash_L \varphi$ whenever $\Gamma \vDash_{I(L)} \varphi$.

Definition 8.4.2 Let L be a modulated logic system.

- L is *globally sound* if $\Gamma \vDash_L^g \varphi$ whenever $\Gamma \vdash_L^g \varphi$ for every Γ and φ in $gL(\Sigma)$ and it is *locally sound* if $\Gamma \vDash_L^\ell \varphi$ whenever $\Gamma \vdash_L^\ell \varphi$ for every Γ and φ in $gL(\Sigma)$.

- L is *globally complete* if $\Gamma \vdash_L^g \varphi$ whenever $\Gamma \vDash_L^g \varphi$ for every Γ and φ in $gL(\Sigma)$ and it is *locally complete* if $\Gamma \vdash_L^\ell \varphi$ whenever $\Gamma \vDash_L^\ell \varphi$ for every Γ and φ in $gL(\Sigma)$. ▽

Next we introduce the notion of modulated logic system morphism. As expected, such morphism is a modulated interpretation system morphism and a modulated Hilbert calculus morphism. However it is also necessary that a particular condition relating both holds. This additional requirement will be referred to as *soundness condition.*

Definition 8.4.3 A *modulated logic system morphism* $h : L \to L'$ is a tuple

$$h = \langle \hat{h}, \underline{h}, \dot{h}, \ddot{h}, \check{h} \rangle$$

where

- $\langle \hat{h}, \underline{h}, \dot{h}, \ddot{h} \rangle$ is a modulated interpretation system morphism from $I(L)$ to $I(L')$;

- $\langle \hat{h}, \check{h} \rangle$ is a modulated Hilbert calculus morphism from $H(L)$ to $H(L')$ such that
$$\ddot{h}_{m'}([\![\varphi']\!]_{m'}) \cong_{\underline{h}(m')} [\![\check{h}(\varphi')]\!]_{\underline{h}(m')}$$
for every $\varphi' \in gL(\Sigma')$ and $m' \in M'$. ▽

Prop/Definition 8.4.4 *Modulated logic systems and their morphisms, with composition and identity maps are defined as expected, constitute the category* **mLog**. ▽

Let Int : **mLog** $\to$ **mInt** and Hil : **mLog** $\to$ **mHil** be the functors that associate to each logic system L the underlying interpretation system $I(L)$ and the underlying Hilbert calculus, $H(L)$. A bridge between two logic systems is now defined as expected using bridges between the interpretation and Hilbert systems involved.

Definition 8.4.5 A *bridge* between logic systems L' and L'' is a diagram
$$\beta = \langle f' : \check{L} \to L', f'' : \check{L} \to L'' \rangle$$
in **mLog** such that

- $Int(\beta) = \langle Int(f') : I(\check{L}) \to I(L'), Int(f'') : I(\check{L}) \to I(L'') \rangle$ is a bridge in **mInt**;
- $Hil(\beta) = \langle Hil(f') : H(\check{L}) \to H(L'), Hil(f'') : H(\check{L}) \to H(L'') \rangle$ is a bridge in **mHil**. ▽

As expected, a bridge $\beta = \langle f' : \check{L} \to L', f'' : \check{L} \to L'' \rangle$ between two logic systems L' and L'' constitutes a bridge between the underlying interpretation systems and a bridge between the underlying Hilbert calculi (see Figure 8.21).

Again modulated fibring between logic systems appears as a pushout in the category of logic systems.

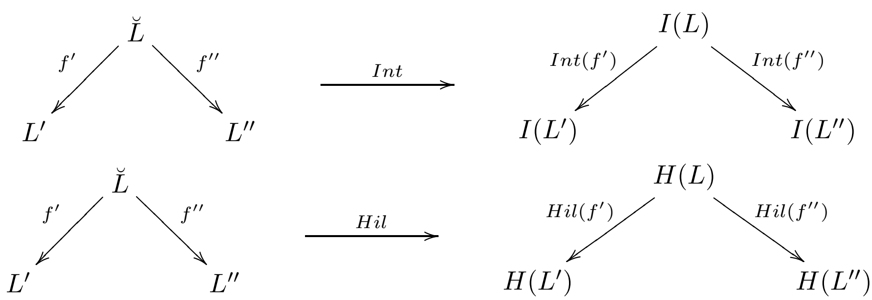


Figure 8.21: Forgetful functors

Prop/Definition 8.4.6 *The* modulated fibring *of logic systems L' and L'' by a bridge $\beta = \langle f' : \breve{L} \to L', f'' : \breve{L} \to L'' \rangle$ is a pushout of β in* **mLog**, *if the pushout exists.*

Proof. The pushout $\langle g' : L' \to L, g'' : L'' \to L \rangle$ (see Figure 8.22) is such that:

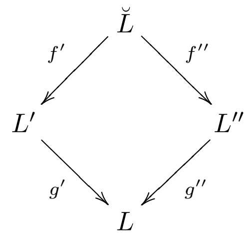


Figure 8.22: Pushout of logic systems

- $\langle Int(g') : I(L') \to I(L), Int(g'') : I(L'') \to I(L) \rangle$ is a modulated fibring in **Int** of $Int(\beta)$;
- $\langle Hil(g') : H(L') \to H(L), Hil(g'') : H(L'') \to H(L) \rangle$ is a modulated fibring in **mHil** of $H(\beta)$.

We show $\ddot{g}''_m(\llbracket \gamma \rrbracket_m) \cong_{\underline{g}''(m)} \llbracket \breve{g}''(\gamma) \rrbracket_{\underline{g}'(m)}$ by induction on the structure of γ. We prove the base when γ is $\hat{g}'(c')$ with $c' \in C'_0$:

$$\begin{aligned} \ddot{g}''_m(\llbracket \hat{g}'(c') \rrbracket_m) &\cong \ddot{g}''_m(\dot{g}'_m(\llbracket c' \rrbracket_{\underline{g}'(m)})) \\ &\cong \dot{f}''_{\underline{f}'(\underline{g}'(m))}(\ddot{f}'_{\underline{g}'(m)}(\llbracket c' \rrbracket_{\underline{g}'(m)})) \\ &\cong \dot{f}''_{\underline{f}''(\underline{g}''(m))}(\llbracket \check{f}'(c') \rrbracket_{\underline{f}''(\underline{g}''(m))}) \\ &\cong \llbracket \hat{f}''(\check{f}'(c')) \rrbracket_{\underline{g}''(m)} \\ &\cong \llbracket \breve{g}''(\gamma) \rrbracket_{\underline{g}''(m)}. \end{aligned}$$

◁

We now present several illustrating examples.

Example 8.4.7 The diagram $\langle f' : \breve{L} \to L', f'' : \breve{L} \to L'' \rangle$ such that

- $\langle \langle \hat{f}', \underline{f}', \dot{f}', \ddot{f}' \rangle : \breve{I}(L) \to I(L'), \langle \hat{f}'', \underline{f}'', \dot{f}'', \ddot{f}'' \rangle : \breve{I}(L) \to I(L'') \rangle$ is the bridge in Proposition 8.2.22;
- $\langle \langle \hat{f}', \check{f}' \rangle : \breve{H}(L) \to H(L'), \langle \hat{f}'', \check{f}'' \rangle : \breve{H}(L) \to H(L'') \rangle$ is the bridge in Proposition 8.3.13;

constitutes a bridge that defines the unconstrained modulated fibring of L' and L''. This happens because the soundness condition is verified: let m' be in M' and $\varphi' \in gL(\Sigma')$, then $\ddot{f}'_{m'}([\![\varphi']\!]_{m'}) = \breve{\top} = [\![\check{f}'(\varphi')]\!]_{\underline{f}'(m')}$. Similarly for f''. ∇

In the Example 8.4.7 we proved that the soundness condition holds when considering the general bridge that can be used to avoid the collapsing when no sharing of symbols is wanted. We give below another example showing that the soundness condition holds. The next example refers to modulated fibring of classical and 3-valued Gödel logics sharing negation.

Example 8.4.8 Consider bridge 2 presented in Example 8.3.10 and the bridge in Example 8.3.16 assuming that $C'_0 = \{\mathbf{f}', \mathbf{t}'\}$ and $C''_0 = \{\mathbf{f}'', \mathbf{t}''\}$. We verify that f' (and similarly for f'') is a logic system morphism. Let m' be in M' and $\varphi' \in gL(\Sigma')$, then,

(i) if $\varphi' = \hat{f}'(\breve{\varphi})$ then

$$\begin{aligned}
\ddot{f}'_{m'}([\![\varphi']\!]_{m'}) &= \ddot{f}'_{m'}([\![\hat{f}'(\breve{\varphi})]\!]_{m'}) \\
&\breve{\cong} \ddot{f}'_{m'}(\dot{f}'_{m'}([\![\breve{\varphi}]\!]_{\underline{f}'(m')})) \\
&\breve{\cong} [\![\breve{\varphi}]\!]_{\underline{f}'(m')} \\
&= [\![\check{f}'(\hat{f}'(\breve{\varphi}))]\!]_{\underline{f}'(m')} \\
&= [\![\check{f}'(\varphi')]\!]_{\underline{f}'(m')};
\end{aligned}$$

(ii) if $\varphi' \vdash^{\ell}_{H'} \mathbf{f}'$ then $\ddot{f}'_{m'}([\![\varphi']\!]_{m'}) = \ddot{f}'_{m'}(\bot') = \breve{\bot} = [\![\breve{\mathbf{f}}]\!]_{\underline{f}'(m')} = [\![\check{f}'(\varphi')]\!]_{\underline{f}'(m')}$ using the fact that L' is sound;

(iii) otherwise $\varphi' \not\vdash^{\ell}_{H'} \mathbf{f}'$ and so $\varphi' \not\vDash^{\ell}_{I'} \mathbf{f}'$ since L' is complete. So there exists m' such that $[\![\varphi']\!]_{m'} \neq \bot'$. Since all constructors in C' have the same denotation in all models then $[\![\varphi']\!]_{m'} \neq \bot'$ for all models m'. Therefore

$$\ddot{f}'([\![\varphi']\!]_{m'}) = \breve{\top}$$

for every m'. Thus, $\ddot{f}'_{m'}([\![\varphi']\!]_{m'}) = \breve{\top} = [\![\breve{\mathbf{t}}]\!]_{\underline{f}'(m')} = [\![\check{f}'(\varphi')]\!]_{\underline{f}'(m')}$. ∇

Now we establish a new way of considering modulated fibring of logic systems that satisfy certain requirements. Later we apply the general result to the modulated fibring of 3-valued Gödel and Łukasiewicz logics. In the following proposition we consider a logic system with equivalence with the usual meaning (see Definition 2.3.27).

Prop/Definition 8.4.9 *Let $L = \langle \Sigma, M, A, R_d, R_p \rangle$ be a sound and complete logic system with equivalence and such that M is countable. Then, the logic system*

$$\tilde{L} = \langle \tilde{\Sigma}, M, \tilde{A}, \tilde{R}_d, \tilde{R}_p \rangle$$

defined as follows

- *$\tilde{\Sigma}$ is equal to Σ except $\tilde{C}_0 = C_0 \cup G$ where G is composed by 0-ary constructors $c_{\vec{b}}$ for all possible $\vec{b}$ where $\vec{b}$ is a sequence $\langle b_1, b_2, \ldots \rangle$ such that $b_i \in B_{m_i}$, assuming that $M = \{m_1, m_2, \ldots\}$;*

- *$\tilde{A}(m_i)$ is equal to $A(m_i)$ except that $\nu_{m_i}(c_{\vec{b}}) = b_i$;*

- *$\tilde{R}_\ell$ includes R_ℓ and*

  – $\{\langle\{c_{\vec{b}}\}, \varphi\rangle, \langle\{\varphi\}, c_{\vec{b}}\rangle \mid$ *for all* $\varphi, c_{\vec{b}}$ *with* $c_{\vec{b}} \vDash^{\ell}_{I(\tilde{L})} \varphi, \varphi \vDash^{\ell}_{I(\tilde{L})} c_{\vec{b}}\}$

  – $\{\langle\{c_{1\vec{b}}\}, c_{2\vec{b}}\rangle \mid$ *for all* $c_{1\vec{b}}, c_{2\vec{b}}$ *such that* $c_{1\vec{b}} \vDash^{\ell}_{I(\tilde{L})} c_{2\vec{b}}\}$

  – $\{\langle\{(\vec{\varphi} \Leftrightarrow \vec{\delta})\}, (c(\vec{\varphi}) \Leftrightarrow c(\vec{\delta}))\rangle \mid$ *for all sequences of formulas* $\vec{\varphi}$ *and* $\vec{\delta}\}$;

- $\tilde{R}_g = \tilde{R}_\ell \cup R_g \cup \{\langle\{c_{1\vec{b}}\}, c_{2\vec{b}}\rangle \mid$ *for all* $c_{1\vec{b}}, c_{2\vec{b}}$ *such that* $c_{1\vec{b}} \vDash^{g}_{I(\tilde{L})} c_{2\vec{b}}\}$

is sound, complete, with congruence and it is a conservative extension of L.

The result in 8.4.9 can be applied to define the modulated fibring of an extension of Gödel logic and Łukasiewicz logic sharing conjunction and disjunction.

Example 8.4.10 Let L' be the logic system for the 3-valued Gödel logic we obtain from the modulated interpretation system and Hilbert calculus in Examples 8.2.5 and 8.3.5 and let L'' be the logic system for the 3-valued Łukasiewicz logic we obtain from the modulated interpretation system and Hilbert calculus in Examples 8.2.6 and 8.3.4. Both logic systems are sound, complete, with equivalence and with a finite set of models (with the same truth values). We also assume that they have the same set of arity 0 constructors besides $\mathbf{t}$ and $\mathbf{f}$. Let G be defined as in Prop/Definition 8.4.9. Consider $\tilde{L}'$ and $\tilde{L}''$, the extensions of L' and L'', respectively. Consider the following bridge:

- $\breve{C}_0 = G \cup \{\mathbf{t}, \mathbf{f}\}$, $\breve{C}_2 = \{\breve{\wedge}, \breve{\vee}\}$, $\breve{C}_k = \emptyset$ when $k \geq 3$ and $k = 1$, $\breve{\&}$ is $\breve{\wedge}$, $\breve{\Xi} = \emptyset$ and $\breve{S} = \emptyset$;

- $\breve{M} = A'(M')|_{\breve{C}} \cup A''(M'')|_{\breve{C}}$;

- $\breve{R}_\ell$ and $\breve{R}_g$ are translations of all ground instances of $\tilde{R}_g{}'$, $\tilde{R}_\ell{}'$, $\tilde{R}_g{}''$, $\tilde{R}_\ell{}''$ by $\check{f}'$ and $\check{f}''$;

- $\hat{f}'$ and $\hat{f}''$ are injections;

- $\check{f}'(\varphi') = \begin{cases} \breve{\varphi} & \text{if } \varphi' \text{ is } \hat{f}'(\breve{\varphi}) \\ c_{\varphi'} & \text{otherwise} \end{cases}$, similarly for $\check{f}''$;

- $\underline{f}'$, $\dot{f}'_{m'}$ and $\ddot{f}'_{m'}$ are identities, similarly for $\underline{f}''$, $\dot{f}''_{m''}$ and $\ddot{f}''_{m''}$.

We show that f' is indeed a morphism.

1. We first prove that $\varphi' \cong^{\ell}_{H(L')} \hat{f}'(\check{f}'(\varphi'))$. If $\varphi' = \hat{f}'(\check{\varphi})$ then $\hat{f}'(\check{f}'(\varphi')) = \varphi'$. Otherwise $\hat{f}'(\check{f}'(\varphi')) = c_{\varphi'}$ and so $\varphi' \cong'_d c_{\varphi'}$.

2. The condition $\check{f}'(\hat{f}'(\check{\varphi})) \vdash^{\ell}_{H(\check{L})} \check{\varphi}$ is straightforward.

3. $\hat{f}'(\check{c})(\hat{f}'(\check{f}'(\varphi'_1)), \hat{f}'(\check{f}'(\varphi'_2))) \vdash^{\ell}_{H(L')} \hat{f}'(\check{c})(\varphi'_1, \varphi'_2)$ because

$$\varphi' \vdash^{\ell}_{H(L')} \hat{f}'(\check{f}'(\varphi'))$$

and L' has congruence. ∇

We synthesize the properties of the modulated fibring of logics in the following way:

- *homogeneous combination mechanism at the deductive level*: both original logics are presented by Hilbert calculi;
- *homogeneous combination mechanism at the semantic level*: both original logics are presented by interpretation structures;
- *algorithmic combination of logics at the deductive level*: given the Hilbert calculi for the original logics, we know how to define the Hilbert calculus for the fibring;
- *semi-algorithmic combination of logics at the semantic level*: given the classes of interpretation structures for the original logics and a (pre-defined) bridge, we know how to define the class of interpretation structures for the modulated fibring, but in many cases the given logics have to be pre-processed (that is, the interpretation structures for the original logics have to be extracted).

8.5 Preservation results

In this section we discuss preservation results. In Subsection 8.5.1 we address the preservation of soundness by modulated fibring. In Subsection 8.5.2 we address the preservation of completeness by modulated fibring.

8.5.1 Soundness

Herein, we address the problem of soundness preservation. The goal is to establish a result stating that if we start with sound modulated logic systems then the modulated logic system obtained by modulated fibring is again sound. To this end, we first provide a sufficient condition for a logic system to be sound and then we establish the conditions for soundness preservation in this setting.

In the sequel, we need the following technical lemma stating that safe substitutions preserve entailment.

Lemma 8.5.1 *Let $I = \langle \Sigma, M, A \rangle$ be a modulated interpretation system, φ a formula, Γ a set of formulas and σ a safe substitution for $\Gamma \cup \{\varphi\}$. Then,*

$$\sigma(\Gamma) \vDash_I^\ell \sigma(\varphi) \text{ and } \sigma(\Gamma) \vDash_I^g \sigma(\varphi)$$

whenever $\Gamma \vDash_I^\ell \varphi$ and $\Gamma \vDash_I^g \varphi$, respectively.

Proof. Observe that

$$[\![\delta]\!]_m^{\alpha_\sigma} = [\![\sigma(\delta)]\!]_m^\alpha$$

where α_σ is an assignment such that $\alpha_\sigma(\xi) = [\![\sigma(\xi)]\!]_m^\alpha$ which can be proved by a straightforward induction. Note also that α_σ is safe for $\Gamma \cup \{\delta\}$ whenever α is safe for $\sigma(\Gamma \cup \{\delta\})$.

(i) Assume that $\Gamma \vDash_I^\ell \delta$. As a consequence, there is a finite set $\Phi \subseteq \Gamma$ such that $\sqcap_{\varphi\in\Phi}[\![\varphi]\!]_m^\alpha \leq [\![\delta]\!]_m^\alpha$ for every model m in M and assignment α safe for $\Phi \cup \{\delta\}$. Let m be a model in M and α an assignment over m safe for $\sigma(\Phi \cup \{\delta\})$. Hence

$$\sqcap_{\varphi\in\Phi}[\![\varphi]\!]_m^{\alpha_\sigma} \leq [\![\delta]\!]_m^{\alpha_\sigma}.$$

Then $\sqcap_{\varphi\in\Phi}[\![\sigma(\varphi)]\!]_m^\alpha \leq [\![\sigma(\delta)]\!]_m^\alpha$. Therefore $\sigma(\Phi) \vDash_I^\ell \sigma(\delta)$ and so $\sigma(\Gamma) \vDash_I^\ell \sigma(\delta)$.

(ii) The proof for $\vDash_I^g$ is analogous. ◁

We now establish the usual sufficient condition for soundness of logic systems: a modulated logic system L is sound whenever its inference rules are sound. We express this condition in terms of models for the modulated Hilbert calculus $H(L)$. We say that a model m is a *model for Hilbert calculus H* if for every rule $\langle \Delta, \delta \rangle \in R_g$, $m\alpha \Vdash_{\mathrm{g}} \delta$ whenever $m\alpha \Vdash_{\mathrm{g}} \gamma$ for every $\gamma \in \Gamma$ and safe assignment α for $\Delta \cup \{\delta\}$ and for every rule $\langle \Delta, \delta \rangle \in R_\ell$, $m\alpha b \Vdash \delta$ whenever $m\alpha b \Vdash \varphi$ for every $\varphi \in \Delta$, safe assignment α for $\Delta \cup \{\delta\}$ and $b \in B_m$.

Proposition 8.5.2 *Let L be a modulated logic system such that each $m \in M$ is a model for $H(L)$. Then L is sound with respect to global and local and derivations.*

Recall that in the logic system L obtained by modulated fibring there are additional rules that are not inherited from the component logic systems. Therefore, to ensure the soundness of L, we have to prove that, besides being a model for each inherited rule, every model in L is also model for these additional rules.

Before establishing the soundness preservation result, we need the next auxiliary lemma.

Lemma 8.5.3 *Let $h : L \to L'$ be a modulated logic system morphism such that $id_\Sigma \in S$.*

1. *$[\![\varphi']\!]_{m'} \cong [\![\hat{h}(\check{h}(\varphi'))]\!]_{m'}$, whenever $\varphi' \in gL(\Sigma', \hat{h})$ and $\ddot{h}_{m'}$ is surjective, for any $m' \in M'$.*
2. *$[\![\varphi']\!]_{m'} \leq' [\![\hat{h}(\check{h}(\varphi'))]\!]_{m'}$, whenever $\varphi' \in gL(\Sigma')$, for any $m' \in M'$.*

Proof.

1. Noting that $\ddot{h}_{m'}$ is surjective,

$$
\begin{aligned}
&[\![\hat{h}(c)(\gamma_1',\dots,\gamma_k')]\!]_{m'}\\
&\qquad \cong \nu_{m'}(\hat{h}(c))([\![\gamma_1']\!]_{m'},\dots,[\![\gamma_k']\!]_{m'})\\
&\qquad \cong \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}([\![\gamma_1']\!]_{m'}),\dots,\ddot{h}_{m'}([\![\gamma_k']\!]_{m'})))\\
&\qquad \cong \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)([\![\check{h}(\gamma_1')]\!]_{\underline{h}(m')},\dots,[\![\check{h}(\gamma_k')]\!]_{\underline{h}(m')}))\\
&\qquad \cong \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}(\dot{h}_{m'}([\![\check{h}(\gamma_1')]\!]_{\underline{h}(m')})),\dots,\ddot{h}_{m'}(\dot{h}_{m'}([\![\check{h}(\gamma_k')]\!]_{\underline{h}(m')}))))\\
&\qquad \cong \dot{h}_{m'}(\nu_{\underline{h}(m')}(c)(\ddot{h}_{m'}([\![\hat{h}(\check{h}(\gamma_1'))]\!]_{m'}),\dots,\ddot{h}_{m'}([\![\hat{h}(\check{h}(\gamma_k'))]\!]_{m'})))\\
&\qquad \cong \nu_{m'}(\hat{h}(c))([\![\hat{h}(\check{h}(\gamma_1'))]\!]_{m'},\dots,[\![\hat{h}(\check{h}(\gamma_k'))]\!]_{m'})\\
&\qquad \cong [\![\hat{h}(c)(\hat{h}(\check{h}(\gamma_1')),\dots,\hat{h}(\check{h}(\gamma_k')))]\!]_{m'}.
\end{aligned}
$$

2. $[\![\varphi']\!]_{m'} \leq' \dot{h}_{m'}(\ddot{h}_{m'}([\![\varphi']\!]^{m'})) \cong \dot{h}_{m'}([\![\check{h}(\varphi')]\!]_{\underline{h}(m')}) \cong [\![\hat{h}(\check{h}(\varphi'))]\!]_{m'}$. ◁

We are now ready to present the main result on preservation of soundness. The logic system obtained by modulated fibring is sound whenever each component logic system is sound and, in each component logic system, the identity signature morphism is one of its safe-relevant morphisms.

Theorem 8.5.4 *The modulated logic system L in the modulated fibring*

$$\langle g' : L' \to L, g'' : L'' \to L\rangle$$

of L' and L'' by a bridge β is sound, provided that L' and L'' are sound, $\mathrm{id}_{\Sigma'} \in S'$ and $\mathrm{id}_{\Sigma''} \in S''$.

Proof. Taking into account Proposition 8.5.2 we just need to check that each model in M is a model for H. Let $r = \langle \Delta, \delta\rangle \in R_\ell$.

(i) Assume r is $\hat{g}'(\langle \Delta', \delta'\rangle)$ with $\langle \Delta', \delta'\rangle$ in $R_\ell{}'$. Then $\Delta \vDash^\ell_L \delta$ by Proposition 8.2.16 since $\ddot{h}_m$ is surjective for each $m \in M$, $\hat{g}'(\Delta') \vDash^\ell_{L'} \hat{g}'(\delta')$ and $\mathrm{id}_{\Sigma'} \in S'$.

(ii) Assume $r = \langle \{\gamma\}, \hat{g}'(\check{g}'(\gamma))\rangle$ with γ in $gL(\Sigma)$. Then, using Lemma 8.5.3, we have

$$[\![\gamma]\!]_m \leq [\![\hat{g}'(\check{g}'(\gamma))]\!]_m$$

for $m \in M$. Hence $\gamma \vDash^\ell_L \hat{g}'(\check{g}'(\gamma))$.

(iii) Assume r is $\langle \{\hat{g}'(c')(\hat{g}'(\check{g}'(\vec{\gamma})))\}, \hat{g}'(c')(\vec{\gamma})\rangle$ with $\vec{\gamma}$ a sequence over $gL(\Sigma)$. Then, by Lemma 8.5.3,

$$[\![\hat{g}'(c')(\hat{g}'(\check{g}'(\vec{\gamma})))]\!]_m \cong [\![\hat{g}'(c')(\vec{\gamma})]\!]_m$$

for each $m \in M$. Hence, $\hat{g}'(c')(\hat{g}'(\check{g}'(\vec{\gamma}))) \vDash^\ell_L \hat{g}'(c')(\vec{\gamma})$.

For $\langle \Delta, \delta\rangle$ in R_g we can conclude that $\Delta \vDash^g_L \delta$ in a similar way. ◁

8.5.2 Completeness

In this section we study the completeness property. Our goal is to establish preservation results for this property. The first main result gives a sufficient condition for completeness of a modulated logic system. This result is presented in Proposition 8.5.10. The second main result provides sufficient conditions for the preservation of completeness by modulated fibring. This result is presented in Proposition 8.5.16.

To deal with completeness issues we adopt the Lindenbaum-Tarski approach. Therefore we have to guarantee that the Hilbert calculi we work with are Hilbert calculi with congruence.

Definition 8.5.5 A modulated Hilbert calculus H is said to be a modulated Hilbert calculus with *congruence* if, for every globally closed set Γ, $c(\vec{\varphi}) \cong_\Gamma c(\vec{\delta})$ whenever $\vec{\varphi} \cong_\Gamma \vec{\delta}$ for every constructor c. A modulated logic system L is a *modulated logic system with congruence* if $H(L)$ is a Hilbert calculus with congruence. $\triangledown$

It is easy to conclude that & is also congruent: assuming that $\Gamma, \varphi_i \vdash^\ell_H \delta_i$ for $i = 1, 2$, since $\Gamma, (\varphi_1 \& \varphi_2) \vdash^\ell_H \varphi_i$ with $i = 1, 2$, we get $\Gamma, (\varphi_1 \& \varphi_2) \vdash^\ell_H \delta_i$ with $i = 1, 2$ and therefore

$$\Gamma, (\varphi_1 \& \varphi_2) \vdash^\ell_H (\delta_1 \& \delta_2).$$

For our purposes, it is convenient to assume that the modulated logic systems we deal with include the special constructor $\mathbf{t}$ of arity 0.

Definition 8.5.6 A modulated Hilbert calculus is *with true* if $\mathbf{t} \in C_0$ and $\vdash^\ell_{H(L)} \mathbf{t}$. A modulated logic system L is said to be *a logic system with true* whenever $H(L)$ is a modulated Hilbert calculus with true and $\nu_m(\mathbf{t}) = \top_m$ for every m in M. $\triangledown$

We are now ready to introduce the Lindenbaum-Tarski algebra for each set of formulas closed under proofs.

Prop/Definition 8.5.7 *A Hilbert calculus H with congruence and true induces, for every globally closed subset Γ of $gL(\Sigma)$, an interpretation structure $\lambda\tau_\Gamma$ over Σ, called the Lindenbaum-Tarski algebra*[1] *for Γ, defined as follows:*

- $B_{\lambda\tau_\Gamma} = gL(\Sigma)$;
- $\varphi \leq_\Gamma \delta$ *if and only if* $\Gamma, \varphi \vdash^\ell_H \delta$;
- $\varphi \sqcap_\Gamma \delta = (\varphi \& \delta)$ *and* $\sqcap_\Gamma \emptyset = \mathbf{t}$;
- $\nu_{\lambda\tau_\Gamma}(c)(\varphi_1, \ldots, \varphi_k) = c(\varphi_1, \ldots, \varphi_k)$.

[1] Usually, the Lindenbaum-Tarski algebra is presented using equivalent classes of formulas because the underlying interpretation structures are partial orders.

When there is no ambiguity with respect to the set of formulas we are considering, we refer to a Lindenbaum-Tarski algebra for a set Γ as $\lambda\tau$.

Lemma 8.5.8 *Let L be a modulated logic system with congruence and true and Γ a globally closed subset of $gL(\Sigma)$.*

1. *$\varphi \cong_\Gamma \mathbf{t}$ if and only if φ is in Γ.*
2. *$[\![\varphi]\!]^\alpha_{\lambda\tau} = \sigma(\varphi)$ where σ is such that $\sigma(\xi) = \alpha(\xi)$.*

Given a modulated logic system L with congruence and true, a globally closed set Γ contained in $gL(\Sigma)$ and a signature morphism $s : \breve{\Sigma} \to \Sigma$ in S then $B_{\lambda\tau_\Gamma}(s)$ is the set $gL(\Sigma, s)$. Note that the Lindenbaum-Tarski algebra validates the rules in the Hilbert system at hand.

Now we have to guarantee that in a modulated logic system, for each globally closed set of formulas Γ, we have a model whose underlying structure is the Lindenbaum-Tarski algebra for Γ.

Definition 8.5.9 A modulated logic system L with congruence and true is *full* if, for every set of formulas Γ globally closed, there is a model m_Γ such that $A(m_\Gamma)$ is isomorphic to the Lindenbaum-Tarski algebra for Γ.

Note that we can enrich the class of models of a modulated interpretation system with one extra model for each globally closed set Γ corresponding to the Lindenbaum-Tarski algebra for Γ.

We are now ready to state the first main result of this section, that is, the result providing a sufficient condition for a modulated logic system to be complete.

Theorem 8.5.10 *Every full modulated logic system L with congruence and true is complete.*

Proof. Let Γ_0 and δ be in $L(C, \&)$.

(i) Assume that $\Gamma_0 \not\vdash^g_L \delta$. Then $\delta \notin \Gamma$ where Γ is the set $\Gamma_0^{\vdash^g_L}$. So $[\![\delta]\!]_{\lambda\tau_\Gamma} = \delta \not\cong \mathbf{t}$ using Lemma 8.5.8. On the other hand

$$[\![\gamma]\!]_{\lambda\tau_\Gamma} = \gamma \cong \mathbf{t}$$

for every γ in Γ. Therefore $\Gamma \not\vDash^g_{\lambda\tau_\Gamma} \delta$. Let m_Γ be the model in M such that $A(m_\Gamma)$ is isomorphic to $\lambda\tau_\Gamma$. Then $\Gamma \not\vDash^g_{A(m_\Gamma)} \delta$ and so $\Gamma \not\vDash^g_L \delta$.

(ii) Assume $\Gamma_0 \vDash^\ell_L \delta$ and let m_Γ be the model in M whose structure is isomorphic to $\lambda\tau_\Gamma$ where Γ is the set $\emptyset^{\vdash^\ell_L}$. Then there is a finite set $\{\gamma_1, \dots, \gamma_k\} \subseteq \Gamma_0$ such that $\{\gamma_1, \dots, \gamma_k\} \vDash^\ell_{A(m_\Gamma)} \delta$. Hence

$$\{\gamma_1, \dots, \gamma_k\} \vDash^\ell_{\lambda\tau_\Gamma} \delta \text{ and } \sqcap_{i=1,\dots,k} [\![\gamma_i]\!]_{\lambda\tau_\Gamma} \leq_\Gamma [\![\delta]\!]_{\lambda\tau_\Gamma}.$$

So, using Lemma 8.5.8, $\sqcap_{i=1,\dots,k}\gamma_i \leq_\Gamma \delta$. Therefore $\{\gamma_1, \dots, \gamma_k\} \vdash^\ell_L \delta$ and so $\Gamma_0 \vdash^\ell_L \delta$. ◁

Next, the main goal is to establish preservation of completeness by modulated fibring under reasonable conditions. According to Theorem 8.5.10 we can conclude that a modulated logic system is complete provided that it is full and with congruence and true. Therefore we prove that congruence and true are preserved by modulated fibring. Moreover we also prove that fullness is preserved by modulated fibring provided that the bridge has additional properties.

Lemma 8.5.11 *Let $h : H \to H'$ be a modulated Hilbert calculus morphism such that $\hat{h}$ is injective and $\check{h}$ is surjective. Then, $\check{h}(\Gamma')$ is a globally closed set of formulas whenever Γ' is a globally closed set of ground formulas.*

Proof. Let φ in $L(\Sigma)$ be such that $\check{h}(\Gamma') \vdash^g_H \varphi$. Then $\hat{h}(\check{h}(\Gamma')) \vdash^g_{H'} \hat{h}(\varphi)$, so $\Gamma' \vdash^g_{H'} \hat{h}(\varphi)$, hence $\hat{h}(\varphi) \in \Gamma'$ and therefore $\varphi \in \check{h}(\Gamma')$ since $\varphi \cong \check{h}(\hat{h}(\varphi))$. ◁

In the sequel, we need to work with the category **St** of structures as well as the category **St**(Σ) of structures over the same signature Σ.

Prop/Definition 8.5.12 *The category* ***St*** *is such that*

- *objects are tuples $\langle \Sigma, B, \leq, \nu \rangle$ where Σ is a modulated signature and $\langle B, \leq, \nu \rangle$ is a modulated interpretation structure over Σ;*
- *morphisms are tuples $\langle \hat{h}, \dot{h}, \ddot{h} \rangle$ where $\hat{h}$ is a modulated signature morphism, $\dot{h} : \langle B, \leq \rangle \to \langle B', \leq' \rangle$ is a monotonic map, $\ddot{h} : \langle B', \leq' \rangle \to \langle B, \leq \rangle$ is a monotonic map preserving finite meets, $\ddot{h}$ is left adjoint to $\dot{h}$ and*

$$\nu'(\hat{h}(c))(\vec{b'}) = \dot{h}(\nu(c)(\ddot{h}(\vec{b'}))).$$

The category ***St***(Σ) *is the fiber of* ***St*** *over Σ, that is, objects are interpretation structures over Σ and morphisms are pairs $\langle \dot{h}, \ddot{h} \rangle$.* ▽

We show that each modulated Hilbert calculus morphism h induces a morphism between the Lindenbaum-Tarski algebra for each globally closed set Γ and the Lindenbaum-Tarski algebra for $\check{h}(\Gamma)$. The conditions in the definition of modulated Hilbert calculus morphism are essential to establish this result and, in fact, they were introduced with this purpose in mind.

Proposition 8.5.13 *Let H and H' be modulated Hilbert calculi with congruence and true and $h : H \to H'$ a morphism such that $\hat{h}$ is injective and $\check{h}$ is surjective. Then*

$$\langle \hat{h}, \dot{h}_{\Gamma'}, \ddot{h}_{\Gamma'} \rangle : \langle \Sigma, \lambda\tau_{\check{h}(\Gamma')} \rangle \to \langle \Sigma', \lambda\tau_{\Gamma'} \rangle$$

is a morphism in ***St*** *where*

$$\dot{h}_{\Gamma'}(\varphi) = \hat{h}(\varphi) \text{ and } \ddot{h}_{\Gamma'}(\varphi') = \check{h}(\varphi')$$

for every globally closed set Γ' over $gL(\Sigma')$.

Theorem 8.5.14 *The modulated fibring $\langle g' : L' \to L, g'' : L'' \to L\rangle$ of logic systems L' and L'' with congruence and true by a bridge β is a logic systems with congruence and true.*

Proof. Let c be a constructor in C_k, $\Gamma, \vec{\delta} \vdash^{\ell}_{H} \vec{\varphi}$ and $\Gamma, \vec{\varphi} \vdash^{\ell}_{H} \vec{\delta}$. Then c is in $\hat{g}'(C'_k)$ or in $\hat{g}''(C''_k)$. Suppose that there exists c' in C'_k with $c = \hat{g}'_k(c')$. Then

$$\check{g}'(\Gamma), \check{g}'(\vec{\delta}) \vdash^{\ell}_{H'} \check{g}'(\vec{\varphi}) \text{ and } \check{g}'(\Gamma), \check{g}'(\vec{\varphi}) \vdash^{\ell}_{H'} \check{g}'(\vec{\delta}).$$

Since L' has congruence then $\check{g}'(\Gamma), c'(\check{g}'(\vec{\delta})) \vdash^{\ell}_{H'} c'(\check{g}'(\vec{\varphi}))$. Thus

$$\hat{g}'(\check{g}'(\Gamma)), \hat{g}'(c'(\check{g}'(\vec{\delta}))) \vdash^{\ell}_{H} \hat{g}'(c'(\check{g}'(\vec{\varphi}))).$$

Moreover,

$$\Gamma, \hat{g}'_k(c')(\hat{g}'(\check{g}'(\vec{\delta}))) \vdash^{\ell}_{H} \hat{g}'_k(c')(\hat{g}'(\check{g}'(\vec{\varphi})))$$

and finally $\Gamma, \hat{g}'_k(c')(\vec{\delta}) \vdash^{\ell}_{H} \hat{g}'_k(c')(\vec{\varphi})$. The proof of preservation of true is straightforward. $\triangleleft$

Observe that the more complex notion of modulated Hilbert system morphism is used to achieve the preservation of congruence without the requirement, used in Chapter 2, of sharing implication and equivalence. Recall that sharing implication and equivalence lead to the unwanted collapse. For the preservation of fullness by modulated fibring we need further constraints on the bridge.

Definition 8.5.15 A bridge $\langle f' : \breve{L} \to L', f'' : \breve{L} \to L''\rangle$ is *adequate* whenever the logic systems L', L'', $\breve{L}$ are full, with congruence and true and

$$\underline{f}'(m'_{\Gamma'}) = m_{\check{f}(\Gamma')} \text{ and } \underline{f}''(m''_{\Gamma''}) = m_{\check{f}(\Gamma'')}$$

for every globally closed sets of ground formulas Γ' and Γ''.

We would like to use Theorem 8.5.10 to conclude that the modulated fibring of full modulated logic systems with congruence and true by an adequate bridge is complete. For this purpose we have to show that the modulated fibring is full. If Γ is a globally closed set, then there is a model

$$\langle \check{g}'(\Gamma), \check{g}''(\Gamma)\rangle \in M$$

such that $A'(\check{g}'(\Gamma))$ is isomorphic to $\lambda\tau_{\check{g}'(\Gamma)}$ and $A''(\check{g}''(\Gamma))$ is isomorphic to $\lambda\tau_{\check{g}''(\Gamma)}$. We show in Proposition 8.5.19 that $A(\langle \check{g}'(\Gamma), \check{g}''(\Gamma)\rangle)$ is isomorphic to $\lambda\tau_{\Gamma}$.

Theorem 8.5.16 *The modulated fibring $\langle g' : L' \to L, g'' : L'' \to L\rangle$ of modulated logic systems L' and L'' by an adequate bridge β is complete.*

Proof. We know, using Theorem 8.5.14, that L is with congruence and true. We also know that, for each set of formulas Γ closed for proof, the model $\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle$ is in M. Using Proposition 8.5.19 we can also conclude that the interpretation structure

$$A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)$$

is isomorphic to $\lambda\tau_\Gamma$. Therefore, L is full and using Theorem 8.5.10, L is complete. ◁

Example 8.5.17 The following modulated fibrings are complete.

- Unconstrained modulated fibring of full modulated logic systems with congruence and true by an adequate bridge. In particular, the unconstrained modulated fibring of full classical and intuitionistic logics is complete. The same holds for the unconstrained modulated fibring of full classical and Łukasiewicz logics.
- The modulated fibring of full classical logic and Gödel logic sharing negation is complete.
- The modulated fibring of full Gödel logic and Łukasiewicz logic sharing conjunction and disjunction is complete. ▽

An "algebraic" version of the completeness result and the preservation of completeness as in [282] can be also be obtained. Of course in this case congruence is not always preserved by modulated fibring. As proved therein, when the logics have implication and equivalence congruence is preserved.

To conclude the section it remains to prove that the algebra obtained by a pushout of the Lindenbaum-Tarski algebras is isomorphic to the Lindenbaum-Tarski algebra for a globally closed set of formulas in the pushout of the modulated signatures. Before presenting the proof, we need a technical lemma. In order to make the notation lighter, we omit the subscripts of $\dot{g}'_{\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle}$, $\dot{g}''_{\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle}$, $\dot{f}'_{\breve{g}'(\Gamma)}$ and $\dot{f}''_{\breve{g}''(\Gamma)}$.

Lemma 8.5.18 *Let $\langle g' : L' \to L, g'' : L'' \to L\rangle$ be the modulated fibring of modulated logic systems L' and L'' by an adequate bridge β.*

1. $\breve{g}'(\dot{g}''(\varphi'')) = \breve{g}'(\hat{g}''(\varphi''))$.
2. $[\![\varphi]\!]_\alpha^{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} = \dot{g}'(\breve{g}'(\varphi\sigma_\alpha))$ *where $\varphi \in L(\Sigma, \hat{g}')$, α is safe for φ and σ_α is such that*
$$\sigma_\alpha(\xi) = \hat{g}'(\varphi')$$
if $\alpha(\xi) = \dot{g}'(\varphi')$; analogously if $\alpha(\xi) = \dot{g}''(\varphi'')$; moreover
$$\sigma_\alpha(\xi) = \hat{g}'(\varphi')\&\hat{g}''(\varphi'')$$
whenever $\alpha(\xi)$ is not in the co-domain of either $\dot{g}'$ or $\dot{g}''$ and is equal to $\dot{g}'(\varphi') \sqcap \dot{g}''(\varphi'')$.

3. $\dot{g}'(\breve{g}'(\varphi)) \leq \dot{g}''(\breve{g}''(\varphi))$ *for any* $\varphi \in gL(\Sigma, \hat{g}')$.

Proof.
1. We have that

$$\begin{aligned} \ddot{g}'(\dot{g}''(\varphi'')) &= \dot{f}'(\ddot{f}''(\varphi'')) \\ &= \dot{f}'(\breve{f}''(\varphi'')) \\ &= \hat{f}'(\breve{f}''(\varphi'')) \\ &= \breve{g}'(\hat{g}''(\varphi'')). \end{aligned}$$

2. We consider two cases.
(i) We prove that $[\![\xi]\!]_\alpha^{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} = \dot{g}'(\breve{g}'(\sigma_\alpha(\xi))$:

$$\begin{aligned} [\![\xi]\!]_\alpha^{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} &= \alpha(\xi) \\ &= \dot{g}'(\varphi') \\ &= \dot{g}'(\breve{g}'(\hat{g}'(\varphi'))) \\ &= \dot{g}'(\breve{g}'(\sigma_\alpha(\xi)). \end{aligned}$$

(ii) We prove that $[\![\varphi]\!]_\alpha^{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} = \dot{g}'(\breve{g}'(\varphi\sigma_\alpha))$ $(\varphi \notin \Xi)$. The proof follows induction.

Base: if φ is $\hat{g'}(c')$ then

$$\begin{aligned} [\![\varphi]\!]_\alpha^{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} &= \dot{g}'(\nu_{\breve{g}'(\Gamma)}(c')) \\ &= \dot{g}'(c') \\ &= \dot{g}'(\breve{g}'(\hat{g}'(c'))) \\ &= \dot{g}'(\breve{g}'(\sigma_\alpha(\varphi))). \end{aligned}$$

The rest of the proof follows straightforwardly.
3. Observe that

$$\begin{aligned} \dot{g}'(\breve{g}'(\varphi)) &= [\![\varphi]\!]^\alpha_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} \\ &\leq [\![\hat{g}''(\breve{g}''(\varphi))]\!]^\alpha_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} \end{aligned}$$

and that

$$\begin{aligned} [\![\hat{g}''(\breve{g}''(\varphi))]\!]^\alpha_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} &= \dot{g}''(\breve{g}''(\hat{g}''(\breve{g}''(\varphi)))) \\ &= \dot{g}''(\breve{g}''(\varphi)). \end{aligned}$$

◁

Proposition 8.5.19 *Let* $\langle g' : L' \to L, g'' : L'' \to L\rangle$ *be the modulated fibring of modulated logic systems* L' *and* L'' *by an adequate bridge* β. *Then* $\lambda\tau$ *is isomorphic to* $A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)$.

Proof. Consider the maps

$\dot{k} : B_{\lambda\tau} \to B_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)}$ such that

– $\dot{k}(\varphi) = \dot{g}'(\breve{g}'(\varphi))$ whenever φ is in $gL(C, \hat{g}')$ (similarly for φ in $L(C, \hat{g}''))$;

– $\dot{k}(\varphi_1 \& \varphi_2) = \dot{k}(\varphi_1) \sqcap \dot{k}(\varphi_2)$.

$\ddot{k} : B_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)} \to B_{\lambda\tau}$ such that

– $\ddot{k}(\dot{g}'(\varphi')) = \hat{g}'(\varphi')$ (similarly for $\ddot{k}(\dot{g}''(\varphi''))$);
– $\ddot{k}(b_1 \sqcap b_2) = \ddot{k}(b_1) \& \ddot{k}(b_2)$ whenever $b_1 \sqcap b_2$ is not in the co-domain of either $\dot{g}'$ or $\dot{g}''$.

1. We first prove that $\dot{k}$ is monotonic, that is, $\dot{k}(\varphi_1) \le \dot{k}(\varphi_2)$ whenever $\varphi_1 \le \varphi_2$.
(i) The case $\varphi_1 \in L(C, \hat{g}')$ and $\varphi_2 \in L(C, \hat{g}')$ is straightforward.
(ii) Suppose that $\varphi_1 \in L(C, \hat{g}')$ and $\varphi_2 \in L(C, \hat{g}'')$. Then, by Lemma 8.5.18,

$$\begin{aligned} \dot{k}(\varphi_1) &= \dot{g}'(\breve{g}'(\varphi_1)) \\ &\le \dot{g}''(\breve{g}''(\varphi_1)) \\ &\le \dot{g}''(\breve{g}''(\varphi_2)) \\ &= \dot{k}(\varphi_2). \end{aligned}$$

(iii) Suppose that φ_1 is in $L(C, \hat{g}')$ and φ_2 is $\varphi_{21} \& \varphi_{22}$, with φ_{2i} in $L(C, \hat{g}^{s_i})$ for $i = 1, 2$ and $s_i \in \{', ''\}$. Then $\varphi_1 \le \varphi_{21}$ and $\varphi_1 \le \varphi_{22}$. From the previous cases we know that

$$\dot{k}(\varphi_1) \le \dot{k}(\varphi_{21}) \text{ and } \dot{k}(\varphi_1) \le \dot{k}(\varphi_{22}).$$

Hence,

$$\begin{aligned} \dot{k}(\varphi_1) &\le \dot{k}(\varphi_{21}) \sqcap \dot{k}(\varphi_{22}) \\ &= \dot{k}(\varphi_{21} \& \varphi_{22}). \end{aligned}$$

2. We now prove that $\ddot{k}$ is monotonic, that is $\ddot{k}(b_1) \le \ddot{k}(b_2)$ whenever $b_1 \le b_2$.
(i) Assume that $b_1 = \dot{g}'(\varphi_1)$ and $b_2 = \dot{g}'(\varphi_2)$. In this case, $\varphi_1 \le' \varphi_2$. Hence $\hat{g}'(\varphi_1) \le \hat{g}'(\varphi_2)$ and therefore $\ddot{k}(b_1) \le \ddot{k}(b_2)$.
(ii) Assume that $b_1 = \dot{g}'(\varphi_1)$ and $b_2 = \dot{g}''(\varphi_2)$. Then there exists a $\breve{\varphi}$ in $L(\breve{C}, \breve{\&})$ such that

$$\begin{aligned} \dot{g}'(\varphi_1) &\le \dot{g}'(\dot{f}'(\breve{\varphi})) \\ &= \dot{g}''(\dot{f}''(\breve{\varphi})) \\ &\le \dot{g}''(\varphi_2). \end{aligned}$$

So, using the previous case, we have

$$\begin{aligned} \ddot{k}(b_1) &\le \ddot{k}(\dot{g}'(\dot{f}'(\breve{\varphi}))) \\ &= \ddot{k}(\dot{g}''(\dot{f}''(\breve{\varphi}))) \\ &\le \ddot{k}(b_2). \end{aligned}$$

(iii) The case where b_1 and b_2 are not in the co-domain of $\dot{g}'$ and $\dot{g}''$ is straightforward.
3. We now prove that $\ddot{k}$ preserves meets. Let $b_1, b_2 \in B_{A(\langle \breve{g}'(\Gamma), \breve{g}''(\Gamma)\rangle)}$ where $b_1 \sqcap b_2$ is $\dot{g}'(\varphi)$.

(i) Suppose that b_1 is $\dot{g}'(\varphi_1)$ and b_2 is $\dot{g}''(\varphi_2)$. Note that $\ddot{g}'(\dot{g}'(\varphi)) = \ddot{g}'(\dot{g}'(\varphi_1)) \sqcap \ddot{g}'(\dot{g}''(\varphi_2))$. Thus,

$$\varphi \cong \varphi_1 \&' \check{g}'(\hat{g}''(\varphi_2))$$

and

$$\check{g}''(\hat{g}'(\varphi)) \cong \check{g}''(\hat{g}'(\varphi_1)) \&'' \varphi_2.$$

Then

$$\begin{aligned} \hat{g}'(\varphi) &\cong_\Gamma \hat{g}'(\varphi) \& \hat{g}''(\check{g}''(\hat{g}'(\varphi))) \\ &\cong_\Gamma \hat{g}'(\varphi_1) \& \hat{g}''(\varphi_2) \end{aligned}$$

and therefore we have $\ddot{k}(b_1 \sqcap b_2) = \ddot{k}(b_1) \& \ddot{k}(b_2)$.

(ii) Suppose thata b_1 is in the co-domain of $\dot{g}'$ (or $\dot{g}''$) and b_2 is not in the co-domain of $\dot{g}'$ or $\dot{g}''$. Then,

$$\begin{aligned} \ddot{k}(b_1 \sqcap b_2) &= \ddot{k}(b_1 \sqcap b_2' \sqcap b_2'') \\ &= \ddot{k}(b_1 \sqcap b_2') \& \ddot{k}(b_2'') \\ &= \ddot{k}(b_1) \& \ddot{k}(b_2') \& \ddot{k}(b_2'') \\ &= \ddot{k}(b_1) \& \ddot{k}(b_2' \sqcap b_2'') \\ &= \ddot{k}(b_1) \& \ddot{k}(b_2). \end{aligned}$$

(iii) The case where b_1 and b_2 are not in the co-domain of $\dot{g}'$ or in the co-domain of $\dot{g}''$ is straightforward.

4. Now we prove that $\ddot{k}$ is a bijection with inverse $\dot{k}$.

(i) Assume that $\ddot{k} \circ \dot{k} \cong \mathrm{id}_{B_{\lambda\tau}}$. Let φ be in $B_{\lambda\tau}$. If φ is in $L(C, \hat{g}')$, then $\ddot{k}(\dot{k}(\varphi)) = \ddot{k}(\dot{g}'(\check{g}'(\varphi))) = \hat{g}'(\check{g}'(\varphi)) \cong \varphi$. If $\varphi_1 \& \varphi_2$ with φ_i in $L(C, \hat{g}^{j_i})$ for $i = 1, 2$ and $j_i \in \{', ''\}$, then

$$\begin{aligned} \ddot{k}(\dot{k}(\varphi_1 \& \varphi_2))) &= \ddot{k}(\dot{k}(\varphi_1) \sqcap \dot{k}(\varphi_2)) \\ &= \ddot{k}(\dot{k}(\varphi_1)) \sqcap \ddot{k}(\dot{k}(\varphi_2)) \\ &\cong \varphi_1 \sqcap \varphi_2 \\ &= \varphi_1 \& \varphi_2. \end{aligned}$$

(ii) Suppose that $\dot{k} \circ \ddot{k} \cong \mathrm{id}_{B_{A(\langle \check{g}'(\Gamma), \check{g}''(\Gamma) \rangle)}}$. Let b be in $B_{A(\langle \check{g}'(\Gamma), \check{g}''(\Gamma) \rangle)}$. If b is $\dot{g}'(\varphi')$, then

$$\begin{aligned} \dot{k}(\ddot{k}(b)) &= \dot{k}(\hat{g}'(\varphi')) \\ &= \dot{g}'(\check{g}'(\hat{g}'(\varphi'))) \\ &\cong \dot{g}'(\varphi') \\ &= b. \end{aligned}$$

If b is not in the co-domain of $\dot{g}'$ or $\dot{g}''$, then b is $b_1 \sqcap b_2$. So, using the previous cases,

$$\begin{aligned} \dot{k}(\ddot{k}(b)) &= \dot{k}(\ddot{k}(b_1 \sqcap b_2)) \\ &= \dot{k}(\ddot{k}(b_1) \sqcap \ddot{k}(b_2)) \\ &= \dot{k}(\ddot{k}(b_1)) \sqcap \dot{k}(\ddot{k}(b_2)) \\ &= b_1 \sqcap b_2 \\ &= b. \end{aligned}$$

5. $\dot{k}(\nu_{\lambda\tau}(c)(\ddot{k}(b_1),\ldots,\ddot{k}(b_k))) = \nu_{A(\langle\breve{g}'(\Gamma),\breve{g}''(\Gamma)\rangle)}(c)(b_1,\ldots,b_k)$ is straightforward.

So $\lambda\tau$ and $A(\langle\breve{g}'(\Gamma),\breve{g}''(\Gamma)\rangle)$ are isomorphic since

$$k \circ h \cong \mathrm{id}_{A(\langle\breve{g}'(\Gamma),\breve{g}''(\Gamma)\rangle)}$$

and

$$h \circ k \cong \mathrm{id}_{\lambda\tau}$$

where $h : A(\langle\breve{g}'(\Gamma),\breve{g}''(\Gamma)\rangle) \to \lambda\tau$ is a morphism in $\mathbf{St}(\Sigma)$ such that $\dot{h} = \ddot{k}$ and $\ddot{h} = \dot{k}$. ◁

8.6 Final remarks

The chapter describes a solution to the collapsing problem in the context of propositional based logics, both for global and local reasoning. At the semantic level, each algebra in the fibring is related to an algebra in each component logic via a bridge. In a sense, at a first sight, modulated fibring is more related to fibring by functions than algebraic fibring, as described in Chapter 3. Indeed, there are maps relating the sets of truth values of each algebra in the fibring with the truth values of the corresponding algebras in the components and vice-versa. Each pair of such maps constitutes an adjunction between orders. Observe that algebraic fibring is a particular case of modulated fibring. Modulated fibring is very close to the notion of fibring by functions namely precisely because of the adjunction that we referred above. The adjunction imposes that we can circulate between sets of truth-values but puts some restrictions when we come back to the starting set.

The Hilbert calculus is also composed by global and local inference rules but substitution is more constrained in order to cope with the changes made at the semantic level. Preservation of completeness is analyzed following a Lindenbaum-Tarski technique. This means that the completeness proof for modulating fibring cannot be extended to non-truth functional logics where congruence and substitution of equivalents are no longer true.

A restriction of modulating fibring is the assumption that the set of designated values is a singleton. Hence, we do not consider some multi-valued logics where the set of designated values has more than one element. Then the modulated fibring construction can be extended to all classes of finitely-many valued logics where more then one distinguished value is needed [137].

The extension of modulated fibring to the contexts of first-order and higher-order based logics is also worthwhile to study. We believe that the extension to first-order based logics will not be too difficult (if we consider powerset algebras with inclusions instead of general ordered algebras), although some care has to be taken with respect to the provisos and the Hilbert calculus in general. However, the extension to higher-order based logics can be more tricky, namely in what concerns the preservation of soundness.

The extension to first-order based logics keeping the algebraic nature of modulated fibring can only be carried out providing that we investigate the fibring of cylindrical algebras (a nice and updated overview can be found at [1]) and abstract away from this concept so that we can have quantifiers of any arity. It seems like an interesting problem to be pursued by itself.

The impact of modulated fibring when we consider other deductive systems is also of interest. Again in this case the impact of provisos (such as, the variable x should be a fresh variable) seems to be the main problem to be tackled. And, of course, a main challenge appears when considering labeled deductive systems. Of course the problem of fibring the algebras of the labels is a challenging one.

Also of interest is to capture the logic of [82] from the modulated fibring of classical propositional logic and intuitionistic logic not sharing any connectives. The modulated fibring, as referred before, does not capture possible interactions between formulas. Hence some enrichment should be done.

Chapter 9

Splitting logics

One of the basic tenets of this book is that it is interesting, useful and relevant to combine different logical systems into new, coherent systems, with the purpose of using them in many types of applications and in refined forms of reasoning. Several examples in previous chapters were devoted to make clear how this can be done, where the complications are lurking, and how to take profit of this compositionality capacity of logics to expand the expressive power of reasoning as a whole. But, if we can compose logics, why not to decompose them?

Fibring, as we had the opportunity to see in previous chapters, is an apt tool able to combine logics creating new and expressive systems. This combination mechanism goes in the direction of synthesis of logics, what we called *splicing logics*.

The other direction, on the analytical verge, is called *splitting logics* As much as composing (multiplying) and decomposing (factoring) numbers are two sides of the same coin, there is no essential distinction between splicing and splitting, though there are important differences with respect to the aims and applications one may have in mind. Splitting, as a process or an operation to produce logics, has been studied independently from splicing; they can be seen as complementary operations, in the sense the former works top-down, while the later works bottom-up. The expressions "splicing logics" and "splitting logics" were introduced in [47]. The noun "splitting" is also used in the literature in a different sense, designating a "logic that splits a class", as, for instance, in [25].

Possible-translations semantics (proposed in [45]) were briefly mentioned in Section 1.4 of Chapter 1. The main goal of this technique is to offer a better interpretation for a given logic by translating its formulas, in several ways, into a class of simpler logics, with known or acceptable semantics. In case the target logics are not presented semantically, the decomposition is called a *possible-translations characterization*. Several logics exist which are not characterizable by finite matrices, while can be characterized by suitable combinations of many-valued logics through possible-translations.

It is relevant here to recall the notion of matrix semantics, introduced by Jan Łukasiewicz and Emile Post. Matrix semantics generalize algebraic semantics, as used in algebraic logic, and constitute a method for assigning semantic meaning for logics, as well as a method for defining logical systems. Since in many cases matrix semantics will work as basic factors in decomposing logics, we give a brief review of this type of semantics.

What is interesting, as shown in details in Section 9.2, is that matrix semantics are particular cases of possible-translations semantics. This has as a consequence the fact that possible-translations semantics will be adequate for any structural deductive system, since the notion of matrix semantics already enjoys this property.

This chapter also analyzes another techniques for splitting logics which are presented by matrix semantics: direct union of matrices and plain fibring, introduced in [60] (see also [94]).

Even if, as we have discussed above, splicing and splitting as procedures for composing and decomposing logics are but two sides of the same coin, there is difference between them in attitudes and expectancy, as much as there is a difference between multiplying prime factors so as to compose a number, and to decompose a number into prime factors: the basic distinction concerns our inputs and the intended output.

If we represent the result of a process of factoring logics as $\mathcal{L} = \mathcal{L}_1 \odot \mathcal{L}_2$, there are mainly two directions to read this equation:

- Splicing direction
  Starting from logics $\mathcal{L}_1$ and $\mathcal{L}_2$ and $\mathcal{L}$ as inputs, then $\mathcal{L}$ is our output and we have a typical case of splicing $\mathcal{L}_1$ and $\mathcal{L}_2$ obtaining $\mathcal{L}$.

  Example 9.3.11, where the underlying operation is plain fibring, is a clear instance of this procedure.

- Splitting direction
  If $\mathcal{L}$ is the input, the equation represents a typical case of splitting $\mathcal{L}$ into factors $\mathcal{L}_1$ and $\mathcal{L}_2$. The factors may be new logics (or new fragments of "old" logics), in which case the process is innovative, or they may be known logics, in which case the process will be establishing, instead of new logics, new relations among $\mathcal{L}$, $\mathcal{L}_1$ and $\mathcal{L}_2$.

  In this last case there will be no essential difference in considering this as splitting or splicing. Example 9.3.10 and Example 9.2.16 are instances of the case where the factors are new logics, whereas Example 9.3.4 is a case of factoring 3-valued Gödel logic into its own fragments.

The idea of splitting and splicing logics was incipient (even if in rudimentary forms) in some procedures for composing or decomposing logics: some constituents of the possible-translations semantics will already be recognized in some variants of the original notion of fibring (the so-called fibring by functions). Some components of the method of plain fibring of matrices can be distinguished in the fibring by functions as well as in the product of matrices introduced by Jan Łukasiewicz in [185] in the investigation of his four-valued modalities.

This chapter is organized as follows: in Section 9.1 a category of propositional based signatures suitable for splitting logics is introduced, as well as the corresponding category of consequence systems. In Section 9.2 the technique known as possible-translations characterization (and its particular case, possible-translations semantics) is analyzed, and some applications are given. In Section 9.3 two methods for combining matrix logics, plain fibring and direct union of matrices, are reviewed. Such methods, defined in the same lines as fibring by functions, are useful for both splitting and splicing matrix logics. Finally, in Section 9.4, we briefly recall the main ideas discussed in this chapter and present some final comments.

The material of this chapter is based on [48] of [10] and [60]. Some results about products of consequence systems are also taken from [31].

9.1 Basic notions

This section briefly describes the basic definitions, notation and facts concerning propositional based signatures and logics that will be used herein. There are two relevant issues to be noted here.

One one hand, the propositional based signatures C do not include propositional symbols in $\mathbb{P}$. This simplified approach is justified because most of the techniques herein presented are oriented to propositional based languages, and then the schema variables are enough.

On the other hand, a more general notion of signature morphism has to be considered. These morphisms are more appropriate for splitting logics (a process characterized by limits) than for splicing logics (a process characterized by colimits, as seen in the previous chapters).

Throughout this chapter, we assume the denumerable set $\Xi = \{\xi_n : n \in \mathbb{N}^+\}$ of schema variables.

Given a propositional based signature C (see Definition 2.1.1), the *domain of* C is the set $|C| = \bigcup_{k\in\mathbb{N}} C_k$. Recall that, given two signatures C' and C'', their disjoint union is a signature

$$C' \oplus C'' = \{(C' \oplus C'')_k\}_{k\in\mathbb{N}}$$

where $(C' \oplus C'')_k$ is a disjoint union of the sets C'_k and C''_k.

Definition 9.1.1 Given $k \in \mathbb{N}$, $L(C)[k]$ is the set of formulas φ such that the set of schema variables occurring in φ is exactly $\{\xi_1, \ldots, \xi_k\}$. ▽

Note that $\xi_1, \ldots, \xi_k$ are the first k symbols in Ξ. Observe also that $L(C)[0]$ is the set of formulas without schema variables; moreover, $L(C)[k] = \emptyset$ for every

$k \geq 2$ whenever $C_k = \emptyset$ for every $k \geq 2$. We write $\varphi(\xi_1, \ldots, \xi_n)$ to indicate that the schema variables occurring in φ are among $\xi_1, \ldots, \xi_n$.

The notion of complexity $l(\varphi)$ of a formula φ is defined as usual, stipulating that $l(\varphi) = 1$ whenever $\varphi \in \Xi \cup C_0$, and

$$l(c(\psi_1, \ldots, \psi_k)) = 1 + l(\psi_1) + \ldots + l(\psi_k)$$

if $c \in C_k$.

Recall form Definition 2.1.6 that every substitution $\sigma{:}\Xi \to L(C)$ over C can be extended to a unique homomorphism $\widehat{\sigma} : L(C) \to L(C)$.

Given $\varphi(\xi_1, \ldots, \xi_n)$ and σ such that $\sigma(\xi_i) = \psi_i$ $(i = 1, \ldots, n)$ then $\widehat{\sigma}(\varphi)$ will be denoted by $\varphi(\psi_1, \ldots, \psi_n)$.

We now describe the category **sSig** of signatures, whose morphisms generalize those from **Sig**.

Definition 9.1.2 Let C and C' be signatures. A *splitting signature morphism* f from C to C', denoted $f : C \to C'$, is a mapping $f : |C| \to L(C')$ such that, if $c \in C_k$ then $f(c) \in L(C')[k]$. ∇

From now on we will say "signature morphism" instead of "splitting signature morphism".

Given a signature morphism $f : C \to C'$, a mapping $\widehat{f} : L(C) \to L(C')$ can be defined as expected:

- $\widehat{f}(\xi) = \xi$ if $\xi \in \Xi$;
- $\widehat{f}(c) = f(c)$ if $c \in C_0$;
- $\widehat{f}(c(\varphi_1, \ldots, \varphi_k)) = f(c)(\widehat{f}(\varphi_1), \ldots, \widehat{f}(\varphi_k))$ if $c \in C_k$, $\varphi_1, \ldots, \varphi_k \in L(C)$ and $k > 0$.

Observe that $f(c)(\widehat{f}(\varphi_1), \ldots, \widehat{f}(\varphi_k))$ is a notation for $\widehat{\sigma}(f(c))$, where σ is the substitution such that $\sigma(\xi_i) = \widehat{f}(\varphi_i)$ for $i = 1, \ldots, k$.

Clearly the extension $\widehat{f}$ of f is unique. Moreover, if f, f' are signature morphisms such that $\widehat{f} = \widehat{f'}$ then $f = f'$. Additionally, the schema variables occurring in φ and in $\widehat{f}(\varphi)$ are the same.

Example 9.1.3 Consider the signatures C and C' defined as follows:

- $C = \{\neg, \Rightarrow\}$;
- $C' = \{\neg, \vee\}$.

Let $f : C \to C'$ be a signature morphism such that

- $f(\neg) = (\neg\, \xi_1)$;
- $f(\Rightarrow) = ((\neg\, \xi_1) \vee \xi_2)$.

Then, for every $\varphi, \psi \in L(C)$,

- $\widehat{f}(\neg\,\varphi) = (\neg\,\widehat{f}(\varphi))$;
- $\widehat{f}(\varphi \Rightarrow \psi) = ((\neg\,\widehat{f}(\varphi)) \vee \widehat{f}(\psi))$.

In particular,

$$\widehat{f}(\neg(\xi_1 \Rightarrow (\neg\,\xi_2))) = (\neg((\neg\,\xi_1) \vee (\neg\,\xi_2))).$$

∇

Definition 9.1.4 Let $f : C \to C'$ and $g : C' \to C''$ be signature morphisms. The *composition* $g \centerdot f$ of f and g is defined to be the signature morphism $g \centerdot f : C \to C''$ given by the mapping $\widehat{g} \circ f : |C| \to L(C'')$. ∇

The following technical lemmas are useful in the sequel. The reader can find the proofs in [31].

Lemma 9.1.5 *Let $f : C \to C'$ and $g : C' \to C''$ be signature morphisms. Then $\widehat{g \centerdot f} = \widehat{g} \circ \widehat{f}$.*

Lemma 9.1.6 *Let $f : C \to C'$ and let $\sigma : \Xi \to L(C)$ be a substitution over C. Then there is a substitution $\sigma' : \Xi \to L(C')$ over C' such that $\widehat{f} \circ \widehat{\sigma} = \widehat{\sigma'} \circ \widehat{f}$.*

We now introduce a new category of signatures suitable for splitting logics.

Prop/Definition 9.1.7 *Propositional based signatures with morphisms as in Definition 9.1.2 constitute the category* ***sSig*** *of splitting signatures, where composition is as in Definition 9.1.4 and, for every signature C, the identity morphism $id_C : C \to C$ is defined by $id_C(c) = c$, for $c \in C_0$, and $id_C(c) = c(\xi_1, \ldots, \xi_k)$, for $c \in C_k$, $k > 0$.* ∇

The next result, taken from [31], is useful for splitting logics.

Proposition 9.1.8 *The category* ***sSig*** *has products of arbitrary small, non-empty families of objects.*

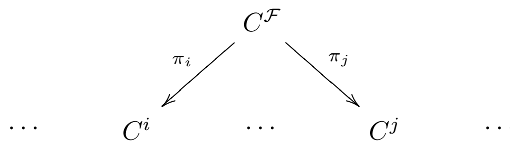
Proof. Let $\mathcal{F} = \{C^i\}_{i \in I}$ be a family of signatures such that I is a non-empty set. Consider the signature $C^{\mathcal{F}}$ such that, for every $k \in \mathbb{N}$,

$$C^{\mathcal{F}}_k = \{(\varphi_i)_{i \in I} : \varphi_i \in L(C^i)[k] \text{ for every } i \in I\}.$$

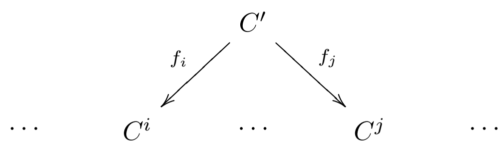
For each $i \in I$, consider the mapping $\pi_i : |C^{\mathcal{F}}| \to L(C^i)$ such that

$$\pi_i((\varphi_i)_{i \in I}) = \varphi_i$$

if $(\varphi_i)_{i \in I} \in C^{\mathcal{F}}_k$, for $k \in \mathbb{N}$. Then π_i determines a **sSig**-morphism $\pi_i : C^{\mathcal{F}} \to C^i$.



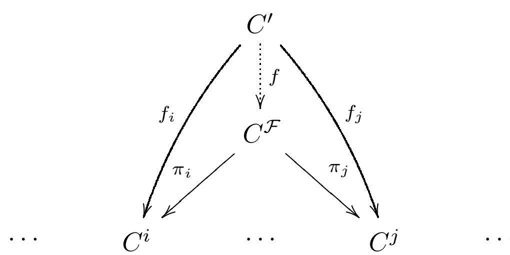
Consider a signature C' together with **sSig**-morphisms $f_i : C' \to C^i$, for $i \in I$.



Let $f : |C'| \to L(C^{\mathcal{F}})$ be the function defined as follows:

$$f(c) = (f_i(c))_{i\in I}(\xi_1, \ldots, \xi_k)$$

if $c \in C'_k$, for $k \in \mathbb{N}$. Then f induces a **sSig**-morphism $f : C' \to C^{\mathcal{F}}$ such that $f_i = \pi_i \cdot f$ for every $i \in I$, that is, the diagram below commutes.



If $g : C' \to C^{\mathcal{F}}$ is a morphism in **sSig** such that $f_i = \pi_i \cdot g$ for every $i \in I$ then clearly $g = f$.

This proves that $\langle C^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ is the product of the family $\mathcal{F}$ in **sSig**. $\triangleleft$

Remark 9.1.9 Let $\mathcal{F}$ be a family in **sSig** as above such that there exist some $i \in I$ with $C^i_k = \emptyset$ for every $k \geq 2$. Then $L(C^i)[k] = \emptyset$ for every $k \geq 2$, and so $C^{\mathcal{F}}_k = \emptyset$ for every $k \geq 2$. On the other hand, if $C^i_0 = \emptyset$ for some $i \in I$ then $C^{\mathcal{F}}_0 = \emptyset$. Note also that $C^{\mathcal{F}}_1 \neq \emptyset$, because $(\xi_1)_{i\in I} \in C^{\mathcal{F}}_1$ for every family $\mathcal{F}$ in **sSig** indexed by a non-empty set I. The signature $C^{\mathcal{F}}$ is an example of a signature containing sets of connectives which (possibly) are not denumerable. $\triangledown$

Example 9.1.10 Let C^1, C^2 and C^3 be signatures such that:

- $\neg_1 \in C^1_1$ and $\Rightarrow_1 \in C^1_2$;
- $\vee_2, \wedge_2 \in C^2_2$;

- $\Box_3 \in C^3_1$ and $\wedge_3 \in C^3_2$.

Let C be the product $C^1 \times C^2 \times C^3$ in **sSig** of C^1, C^2 and C^3. Then

- $c_1 = \langle(\neg_1(\xi_1 \Rightarrow_1 \xi_1)),\ \xi_1,\ (\Box_3(\Box_3\xi_1))\rangle$ is a connective in C_1;
- $c_2 = \langle(\neg_1(\xi_1 \Rightarrow_1 \xi_2)),\ (\xi_2 \vee_2 (\xi_1 \wedge_2 \xi_2)),\ (\xi_2 \wedge_3 (\Box_3\xi_1))\rangle$ is a connective in C_2.

Therefore, $c_2(\xi_1, c_1(\xi_2))$ is a formula in $L(C)[2]$. ∇

Recall consequence systems from Chapter 1. Next we introduce some useful notions about consequence relations.

Definition 9.1.11 Let $\mathcal{C} = \langle C, \vdash\rangle$ be a consequence system and let $C' \leq C$. The *C'-fragment of $\mathcal{C}$* is the consequence system

$$\mathcal{C}|_{C'} = \langle C', \vdash'\rangle$$

where $\vdash' = \vdash \cap\ (\wp(L(C')) \times L(C'))$. ∇

It is worth noting that, $\mathcal{C}' = \mathcal{C}|_{C'}$ if and only if, for every $\Gamma \cup \{\varphi\} \subseteq L(C')$,

$$\Gamma \vdash' \varphi \ \text{ if and only if } \ \Gamma \vdash \varphi.$$

Definition 9.1.12 Let $\mathcal{C} = \langle C, \vdash\rangle$ and $\mathcal{C}' = \langle C', \vdash'\rangle$ be consequence systems.

- $\mathcal{C}$ is a *strong extension* of $\mathcal{C}'$ if $C' \leq C$ and $\vdash' \subseteq \vdash$.
- $\mathcal{C}$ is a *weak extension* of $\mathcal{C}'$ if $C' \leq C$ and

  $$\vdash' \varphi \ \text{ implies that } \ \vdash \varphi$$

  for every $\varphi \in L(C')$. ∇

Note that a consequence system $\mathcal{C}$ is a strong extension of $\mathcal{C}'$ if and only if $\mathcal{C}'$ is weaker than $\mathcal{C}$ (recall Definition 1.1.6 in Chapter 1.

Definition 9.1.13 Let $\mathcal{C} = \langle C, \vdash\rangle$ and $\mathcal{C}' = \langle C', \vdash'\rangle$ be consequence systems.

- $\mathcal{C}$ is a *conservative extension* of $\mathcal{C}'$ if $C' \leq C$ and $\mathcal{C}' = \mathcal{C}|_{C'}$.
- $\mathcal{C}$ is a *conservative weak extension* of $\mathcal{C}'$ if $C' \leq C$ and

  $$\vdash' \varphi \ \text{ if } \ \vdash \varphi$$

  for every $\varphi \in L(C')$. ∇

From the definitions above the following useful result is straightforward; details of the proof are left to the reader.

Recall the notions of structural and standard consequence systems given in Chapter 1.

Proposition 9.1.14

1. *Each fragment over C of any (structural, finitary, standard) consequence system is also a (structural, finitary, standard) consequence system.*
2. *Every consequence system is a conservative extension of any of its fragments over C.*

A new category of consequence systems can now be defined: the category **sCon** which differs from the one introduced in Chapter 1 at the level of morphisms.

Prop/Definition 9.1.15 *The category **sCon** of splitting consequence systems is defined as follows:*

- *the objects are consequence systems;*
- *a morphism $f : \mathcal{C} \to \mathcal{C}'$ in **sCon** is a morphism $f : C \to C'$ in **sSig** such that, for every $\Gamma \cup \{\varphi\} \subseteq L(C)$,*

$$\Gamma \vdash \varphi \;\; \textit{implies} \;\; \widehat{f}(\Gamma) \vdash' \widehat{f}(\varphi);$$

- *composition and identity morphisms are as in **sSig**.* ▽

A fundamental property of **sCon** is the following.

Proposition 9.1.16 *The category **sCon** has products of arbitrary small, non-empty families of objects. Moreover, if every object of the family is structural, so is the product.*

Proof. Let $\mathcal{F} = \{\mathcal{C}_i\}_{i\in I}$ be a family of consequence systems, where I is a non-empty set and $\mathcal{C}_i = \langle C^i, \vdash_i\rangle$ for $i \in I$. Let $\langle C^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ be the product of $\{C^i\}_{i\in I}$ in the category **sSig** (see Proposition 9.1.8), and let

$$\vdash_{\mathcal{F}} \subseteq \wp(L(C^{\mathcal{F}})) \times L(C^{\mathcal{F}})$$

be the relation defined as follows:

$$\Gamma \vdash_{\mathcal{F}} \varphi \;\; \text{if and only if} \;\; \widehat{\pi}_i(\Gamma) \vdash_i \widehat{\pi}_i(\varphi) \text{ for every } i \in I.$$

Let $\mathcal{C}^{\mathcal{F}} = \langle C^{\mathcal{F}}, \vdash_{\mathcal{F}}\rangle$. We claim that the pair $\langle \mathcal{C}^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ is the product in the category **sCon** of the family $\mathcal{F}$. We first prove that $\mathcal{C}^{\mathcal{F}}$ is indeed a consequence system (recall Chapter 1).

(i) $\vdash_{\mathcal{F}}$ is extensional
Let $\Gamma \subseteq L(C^{\mathcal{F}})$, and suppose that $\varphi \in \Gamma$. Then $\widehat{\pi}_i(\varphi) \in \widehat{\pi}_i(\Gamma)$ and so, since $\mathcal{C}_i$ satisfies extensivity, it follows that

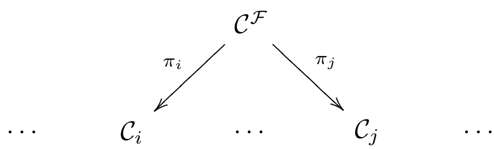
$$\widehat{\pi}_i(\Gamma) \vdash_i \widehat{\pi}_i(\varphi)$$

for every $i \in I$. Therefore $\Gamma \vdash_{\mathcal{F}} \varphi$.
(ii) $\vdash_{\mathcal{F}}$ is transitive
Suppose that $\Gamma \vdash_{\mathcal{F}} \varphi$ and $\Theta \vdash_{\mathcal{F}} \psi$ for every $\psi \in \Gamma$. Fix $i \in I$. Then

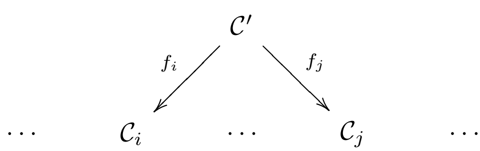
$$\widehat{\pi}_i(\Gamma) \vdash_i \widehat{\pi}_i(\varphi) \text{ and } \widehat{\pi}_i(\Theta) \vdash_i \widehat{\pi}_i(\psi) \text{ for every } \psi \in \Gamma.$$

Thus $\widehat{\pi}_i(\Theta) \vdash_i \widehat{\pi}_i(\varphi)$, because $\vdash_i$ satisfies transitivity, for every $i \in I$. Therefore $\Theta \vdash_{\mathcal{F}} \varphi$.

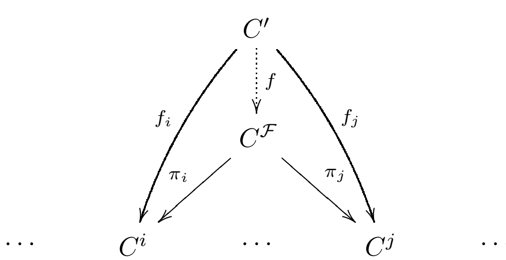
This shows that $\mathcal{C}^{\mathcal{F}}$ is a consequence system. Clearly, each π_i is a **sCon**-morphism $\pi_i : \mathcal{C}^{\mathcal{F}} \to \mathcal{C}_i$, by the very definition of $\mathcal{C}^{\mathcal{F}}$.



Suppose now that $\mathcal{C}' = \langle C', \vdash' \rangle$ is a consequence system and $f_i : \mathcal{C}' \to \mathcal{C}_i$ is a **sCon**-morphism, for every $i \in I$.



Then there exists a unique **sSig**-morphism $f : C' \to C^{\mathcal{F}}$ such that, in **sSig**, $\pi_i \centerdot f = f_i$, for every $i \in I$, because $\langle C^{\mathcal{F}}, \{\pi_i\}_{i \in I} \rangle$ is the product of $\{C^i\}_{i \in I}$ in the category **sSig**. That is, the diagram below commutes in **sSig**.



Suppose that $\Gamma \cup \{\varphi\} \subseteq L(C')$ is such that $\Gamma \vdash' \varphi$. Since each f_i is a **sCon**-morphism then $\widehat{f}_i(\Gamma) \vdash_i \widehat{f}_i(\varphi)$, for every $i \in I$, hence

$$\widehat{\pi_i \centerdot f}(\Gamma) \vdash_i \widehat{\pi_i \centerdot f}(\varphi).$$

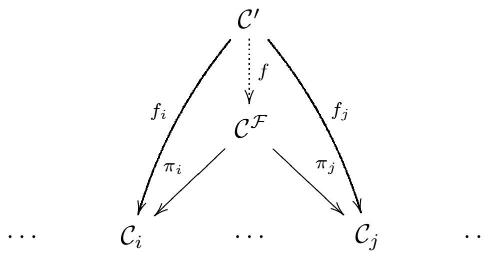
Using the Lemma 9.1.5, we have that

$$\widehat{\pi_i} \circ \widehat{f}(\Gamma) \vdash_i \widehat{\pi_i} \circ \widehat{f}(\varphi)$$

that is,

$$\widehat{\pi_i}(\widehat{f}(\Gamma)) \vdash_i \widehat{\pi_i}(\widehat{f}(\varphi))$$

for every $i \in I$. Therefore, by definition of $\vdash_{\mathcal{F}}$ we have that $\widehat{f}(\Gamma) \vdash_{\mathcal{F}} \widehat{f}(\varphi)$ and then f is a **sCon**-morphism $f : \mathcal{C}' \to \mathcal{C}^{\mathcal{F}}$ such that, in **sCon**, $\pi_i \centerdot f = f_i$, for every $i \in I$. That is, the following diagram is commutative in **sCon**.



The uniqueness of f is a consequence of the universal property in the category **sSig** of the product $\langle C^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$. This show that $\langle \mathcal{C}^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ is the product in the category **sCon** of the family $\mathcal{F}$.

Finally, suppose that every consequence system in $\mathcal{F}$ is structural. We now prove that $\mathcal{C}^{\mathcal{F}}$ is structural. Thus, consider a set $\Gamma \cup \{\varphi\} \subseteq L(C^{\mathcal{F}})$ such that $\Gamma \vdash_{\mathcal{F}} \varphi$. Then, $\widehat{\pi_i}(\Gamma) \vdash_i \widehat{\pi_i}(\varphi)$ for every $i \in I$. Let $\sigma : \Xi \to L(C^{\mathcal{F}})$ be a substitution over $C^{\mathcal{F}}$. Since every π_i is a **sSig**-morphism then, for every $i \in I$, there exists a substitution $\sigma_i : \Xi \to L(C^i)$ over C^i such that $\widehat{\pi_i} \circ \widehat{\sigma} = \widehat{\sigma_i} \circ \widehat{\pi_i}$, by Lemma 9.1.6. Since each $\mathcal{C}_i$ satisfies structurality then

$$\widehat{\sigma_i}(\widehat{\pi_i}(\Gamma)) \vdash_i \widehat{\sigma_i}(\widehat{\pi_i}(\varphi))$$

that is,

$$\widehat{\pi_i}(\widehat{\sigma}(\Gamma)) \vdash_i \widehat{\pi_i}(\widehat{\sigma}(\varphi))$$

for every $i \in I$. Therefore $\widehat{\sigma}(\Gamma) \vdash_{\mathcal{F}} \widehat{\sigma}(\varphi)$ and so $\vdash_{\mathcal{F}}$ satisfies structurality. ◁

A similar result can be proved for compact consequence systems. Note that the product of a (small and non-empty) family $\mathcal{F}$ of compact objects in **sCon**, as defined in the proof of Proposition 9.1.17, may not be a compact consequence system. Thus, a finer tuning is necessary in the definition of $C^{\mathcal{F}}$ in order to obtain a compact object in **sCon**. However, the only possible choice for the appropriate $C^{\mathcal{F}}$ does not satisfy the universal property of the product in **sCon**, when we test the property with consequence systems that are not compact. Thus, it necessary to restrict the category **sCon**. Let **csCon** be the subcategory of **sCon** formed by the compact splitting consequence systems, and with the same morphisms as in **sCon**. That is, **csCon** is a full subcategory of **sCon**. Then, the appropriate restriction of Proposition 9.1.16 to compact consequence systems is as follows.

Proposition 9.1.17 *The category* **csCon** *has products of arbitrary small, non-empty families of objects. Moreover, if every object of the family is structural, so is the product.*

Proof. The proof follows the same lines as that of Proposition 9.1.16. Thus, let $\mathcal{F} = \{\mathcal{C}_i\}_{i\in I}$ be a family of compact consequence systems, where I is a non-empty set and each $\mathcal{C}_i$ is of form $\langle C^i, \vdash_i\rangle$. Consider the product $\langle C^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ of $\{C^i\}_{i\in I}$ in the category **sSig** (see Proposition 9.1.8). Let $\vdash_{\mathcal{F}} \subseteq \wp(L(C^{\mathcal{F}})) \times L(C^{\mathcal{F}})$ be the relation defined as follows:

$\Gamma \vdash_{\mathcal{F}} \varphi$ if and only if
there exists a finite set $\Delta \subseteq \Gamma$ such that $\widehat{\pi}_i(\Delta) \vdash_i \widehat{\pi}_i(\varphi)$ for every $i \in I$.

Let $\mathcal{C}^{\mathcal{F}} = \langle C^{\mathcal{F}}, \vdash_{\mathcal{F}}\rangle$. We will show that the pair $\langle \mathcal{C}^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ is the product in the category **csCon** of the family $\mathcal{F}$. We begin by proving that $\mathcal{C}^{\mathcal{F}}$ is a compact consequence system.

(i) $\vdash_{\mathcal{F}}$ is extensional:
Consider $\Gamma \subseteq L(C^{\mathcal{F}})$. Let $\varphi \in \Gamma$ and $\Delta = \{\varphi\}$. Then $\varphi \in \Delta$ and Δ is a finite subset of Γ. Since $\widehat{\pi}_i(\varphi) \in \widehat{\pi}_i(\Delta)$ and $\mathcal{C}_i$ satisfies extensivity, it follows that $\widehat{\pi}_i(\Delta) \vdash_i \widehat{\pi}_i(\varphi)$, for every $i \in I$. But this means that $\Gamma \vdash_{\mathcal{F}} \varphi$.

(ii) $\vdash_{\mathcal{F}}$ is transitive:
Suppose that $\Gamma \vdash_{\mathcal{F}} \varphi$ and $\Theta \vdash_{\mathcal{F}} \psi$ for every $\psi \in \Gamma$. Then there exists a finite subset $\Delta = \{\gamma_1, \ldots, \gamma_n\}$ of Γ such that, for every $i \in I$, $\widehat{\pi}_i(\Delta) \vdash_i \widehat{\pi}_i(\varphi)$. Let $1 \leq j \leq n$. Then $\Theta \vdash_{\mathcal{F}} \gamma_j$ and so there exists a finite subset Δ_j of Θ such that, for every $i \in I$, $\widehat{\pi}_i(\Delta_j) \vdash_i \widehat{\pi}_i(\gamma_j)$. Let $\Delta' = \bigcup_{j=1}^{n} \Delta_j$. Then Δ' is a finite subset of Θ such that

$$\widehat{\pi}_i(\Delta') \vdash_i \psi$$

for every $\psi \in \widehat{\pi}_i(\Delta_j)$, every $j = 1, \ldots, n$ and every $i \in I$, because every $\vdash_i$ satisfies extensivity. Since every $\vdash_i$ satisfies transitivity then $\widehat{\pi}_i(\Delta') \vdash_i \widehat{\pi}_i(\gamma_j)$, for every $j = 1, \ldots, n$ and every $i \in I$. Using again the transitivity of $\vdash_i$ we infer that

$$\widehat{\pi}_i(\Delta') \vdash_i \widehat{\pi}_i(\varphi)$$

for every $i \in I$. Therefore $\Theta \vdash_{\mathcal{F}} \varphi$.

This shows that $\mathcal{C}^{\mathcal{F}}$ is a consequence system. Moreover, $\mathcal{C}^{\mathcal{F}}$ is compact by the very definition of $\vdash_{\mathcal{F}}$. Clearly, each π_i is a **csCon**-morphism $\pi_i : \mathcal{C}^{\mathcal{F}} \to \mathcal{C}_i$.

Suppose that $\mathcal{C}' = \langle C', \vdash'\rangle$ is a compact consequence system and $f_i : \mathcal{C}' \to \mathcal{C}_i$ is a **csCon**-morphism, for every $i \in I$. Then there exists a unique **sSig**-morphism

$$f : C' \to C^{\mathcal{F}}$$

such that, in **sSig**, $\pi_i \centerdot f = f_i$, for every $i \in I$, because $\langle C^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ is the product of $\{C^i\}_{i\in I}$ in the category **sSig**. Suppose that $\Gamma \cup \{\varphi\} \subseteq L(C')$ is such that $\Gamma \vdash' \varphi$. Since $\mathcal{C}'$ is compact, there exists a finite set $\Delta \subseteq \Gamma$ such that $\Delta \vdash' \varphi$. Since each f_i is a **csCon**-morphism then $\widehat{f}_i(\Delta) \vdash_i \widehat{f}_i(\varphi)$, for every $i \in I$, hence

$$\widehat{\pi_i \centerdot f}(\Delta) \vdash_i \widehat{\pi_i \centerdot f}(\varphi).$$

Using the Lemma 9.1.5, we have that

$$\widehat{\pi}_i \circ \widehat{f}(\Delta) \vdash_i \widehat{\pi}_i \circ \widehat{f}(\varphi)$$

that is,

$$\widehat{\pi}_i(\widehat{f}(\Delta)) \vdash_i \widehat{\pi}_i(\widehat{f}(\varphi))$$

for every $i \in I$, where $\widehat{f}(\Delta)$ is a finite subset of $\widehat{f}(\Gamma)$. Therefore, by definition of $\vdash_{\mathcal{F}}$ we have that $\widehat{f}(\Gamma) \vdash_{\mathcal{F}} \widehat{f}(\varphi)$ and then f is a **csCon**-morphism $f : \mathcal{C}' \to \mathcal{C}^{\mathcal{F}}$ such that, in **csCon**, $\pi_i \centerdot f = f_i$, for every $i \in I$. The uniqueness of f is a consequence of the universal property in the category **sSig** of the product $\langle C^{\mathcal{F}}, \{\pi_i\}_{i \in I}\rangle$. This show that

$$\langle \mathcal{C}^{\mathcal{F}}, \{\pi_i\}_{i \in I}\rangle$$

is the product in the category **csCon** of the family $\mathcal{F}$.

Finally, suppose that every consequence system in $\mathcal{F}$ is structural. We will show that $\mathcal{C}^{\mathcal{F}}$ is structural. Thus, consider a set $\Gamma \cup \{\varphi\} \subseteq L(C^{\mathcal{F}})$ such that $\Gamma \vdash_{\mathcal{F}} \varphi$. Then, there is a finite set $\Delta \subseteq \Gamma$ such that

$$\widehat{\pi}_i(\Delta) \vdash_i \widehat{\pi}_i(\varphi)$$

for every $i \in I$. Let $\sigma : \Xi \to L(C^{\mathcal{F}})$ be a substitution over $C^{\mathcal{F}}$. Since every π_i is a **sSig**-morphism then, for every $i \in I$, there exists a substitution

$$\sigma_i : \Xi \to L(C^i)$$

over C^i such that $\widehat{\pi}_i \circ \widehat{\sigma} = \widehat{\sigma}_i \circ \widehat{\pi}_i$, by Lemma 9.1.6. Since each $\mathcal{C}_i$ satisfies structurality then

$$\widehat{\sigma}_i(\widehat{\pi}_i(\Delta)) \vdash_i \widehat{\sigma}_i(\widehat{\pi}_i(\varphi))$$

that is,

$$\widehat{\pi}_i(\widehat{\sigma}(\Delta)) \vdash_i \widehat{\pi}_i(\widehat{\sigma}(\varphi))$$

for every $i \in I$, where $\widehat{\sigma}(\Delta)$ is a finite subset of $\widehat{\sigma}(\Gamma)$. Therefore $\widehat{\sigma}(\Gamma) \vdash_{\mathcal{F}} \widehat{\sigma}(\varphi)$ and so $\vdash_{\mathcal{F}}$ satisfies structurality.

$\triangleleft$

9.2 Possible-translations semantics

Recall the notions of possible-translations briefly summarized in Section 1.4 of Chapter 1. The present section describes them in more detail and outlines a categorical rendering originally propounded in [30].

The concept of possible-translations characterization (PTC), and in particular of possible-translations semantics (PTS), is based on the idea of defining a new global consequence relation by combining other, presumably simpler, consequence relations by means of translations. In this way, as commented in Chapter 1, PTCs can be seen as a splitting procedure.

The intuition behind possible–translations characterization is to encompass several basic logic systems (perhaps presented semantically, in which case we refer to them as a possible-translations semantics), in such a way as to define a new logic which depends upon the basic ones by means of a collection of translations. The basic logics can be copies of a same logic, or a bunch of logics of distinct nature, as for example distinct many–valued logics, or even modal logics interpreted by Kripke models.

In [47] a somewhat more abstract account of possible-translations semantics was investigated, considering the basic models as organized through sheaf structures. As is well known, sheaves are used in mathematics as a tool for investigating the relationship between local and global phenomena, and seems to be an adequate framework to frame the idea of possible translations.

As mentioned above, instead of thinking of synthesizing some given logics through a combination process in order to obtain a new logic (as is done with fibring, for instance), a logic can be split into a family of other logics; this question can be examined in terms of categories, resulting in a universal construction. Recall from Definition 1.4.1 of Chapter 1 the notions of translation, weak translation and conservative translation between consequence systems. It is convenient to introduce the following concept of conservative morphism.

Definition 9.2.1 Let $\mathcal{C}_i = \langle C^i, \vdash_i \rangle$ for $i = 1, 2$ be consequence systems. A morphism $f : \mathcal{C}_1 \to \mathcal{C}_2$ in **sCon** is said to be *conservative* if

$$\widehat{f} : L(C^1) \to L(C^2)$$

is a conservative translation. ▽

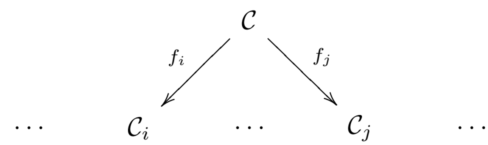
Observe that each morphism f in **sCon** induces a translation $\widehat{f}$ between consequence systems in the sense of the Definition 1.4.1; we call it a *grammatical translation*, in the sense that each k-ary connective c is mapped by f into a formula $\varphi_c(\xi_1, \ldots, \xi_k)$ which uses exactly the schema variables $\xi_1, \ldots, \xi_k$. This formula should be seen as kind of generalized k-ary connective. In Examples 9.2.16 and 9.2.21 some non-grammatical translations will be presented.

We begin by adapting the original definitions of [46] in order to make them suitable for categorical formalization. Some of these definitions were already presented in Chapter 1.

Definition 9.2.2 Let $\mathcal{C} = \langle C, \vdash \rangle$ be a consequence system, and let $\{\mathcal{C}_i\}_{i \in I}$ be a family of consequence systems indexed by a class I such that $\mathcal{C}_i = \langle C^i, \vdash_i \rangle$ for every $i \in I$. A *possible-translations frame for* $\mathcal{C}$ is a pair

$$P = \langle \{\mathcal{C}_i\}_{i \in I}, \{f_i\}_{i \in I} \rangle$$

such that $f_i : L(C) \to L(C^i)$ is a translation between $\mathcal{C}$ and $\mathcal{C}_i$, for every $i \in I$.



A frame $P = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$ is said to be:

- *small* if the class I is a set:
- *grammatical* if f_i is a morphism $f_i : \mathcal{C} \to \mathcal{C}_i$ in **sCon**, for every $i \in I$;
- *structural* if $\mathcal{C}_i$ is structural for every $i \in I$;
- *compact* if $\mathcal{C}_i$ is compact for every $i \in I$. $\triangledown$

Definition 9.2.3 Let $P = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$ be a possible-translations frame for a consequence system $\mathcal{C} = \langle C, \vdash\rangle$. We say that P is a *possible-translations characterization for* $\mathcal{C}$ (in short, a PTC) if, for every $\Gamma \cup \{\varphi\} \subseteq L(C)$,

$$\Gamma \vdash \varphi \;\text{ if and only if }\; f_i(\Gamma) \vdash_i f_i(\varphi) \;\text{ for every } i \in I.$$

In case every system $\mathcal{C}_i$ is presented by semantic means we say that P is a possible-translations semantics for $\mathcal{C}$ (in short, a PTS). A possible-translations characterization P is said to be *small* (respectively, *grammatical*, *structural*, *compact*) if it is small (respectively, grammatical, structural, compact) regarded as a frame. Of course the same qualifications apply to possible-translations semantics. $\triangledown$

In order to obtain a categorical representation of PTCs (see Propositions 9.2.5 and 9.2.6 below), possible-translations frames must here be restricted to small grammatical ones.

As mentioned above, a PTC for a consequence system $\mathcal{C}$ can be seen as a way of splitting the consequence system $\mathcal{C}$ into the family $\{\mathcal{C}_i\}_{i\in I}$ of consequence systems through the translations $\{f_i\}_{i\in I}$.

It is also interesting to consider a notion of PTC weaker than the one in Definition 9.2.3.

Definition 9.2.4 Let $\mathcal{C} = \langle C, \vdash\rangle$ be a consequence system, and let $\{\mathcal{C}_i\}_{i\in I}$ be a family of consequence systems indexed by a class I such that $\mathcal{C}_i = \langle C^i, \vdash_i\rangle$ for every $i \in I$. A *weak possible-translations characterization for* $\mathcal{C}$ is a pair

$$P = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$$

such that, for every $\varphi \in L(C)$,

$$\vdash \varphi \;\text{ if and only if }\; \vdash_i f_i(\varphi) \;\text{ for every } i \in I.$$

In case the consequence systems possess a semantical nature we say that P is a *weak possible-translations semantics.* ∇

Actually, the notion of (weak) possible-translations characterization and its particularization, the (weak) possible-translations semantics, constitute a common abstraction of several inter-relations between distinct logics found in the literature. Besides the example of rendering provability logic in terms of multiple translations into Peano arithmetic (see Example 1.4.5 of Chapter 1), Example 9.2.15 below obtains a (weak) possible-translations semantics for propositional intuitionistic logic in terms of the variety of Heyting algebras, where all translations are the identity mapping. Other examples are Gödel and Gentzen's so-called *negative translation* between classical and intuitionistic logic (see [127, 122, 123]), as well as Gödel's (weak and conservative) translation between intuitionistic and modal logic (see [126]).

The negative translation of a formula φ of first-order classical logic proposed in [127, 122, 123] is the formula $f(\varphi)$ of first-order intuitionistic logic inductively defined according to the following clauses:

- $f(p)$ is $(\neg(\neg p))$, if p is atomic;
- $f(\neg\varphi)$ is $(\neg f(\varphi))$;
- $f(\varphi \wedge \psi)$ is $(f(\varphi) \wedge f(\psi))$;
- $f(\varphi \vee \psi)$ is $(\neg((\neg f(\varphi)) \wedge (\neg f(\psi))))$;
- $f(\varphi \Rightarrow \psi)$ is $(f(\varphi) \Rightarrow f(\psi))$;
- $f(\forall x(\varphi))$ is $(\forall x(f(\varphi)))$;
- $f(\exists x(\varphi))$ is $(\neg(\forall x(\neg f(\varphi))))$.

This translation, independently discovered by Gerhard Gentzen and Kurt Gödel, and anticipated by Andrey Kolmogorov in [164], provides a weak and conservative translation from first-order classical logic into first-order intuitionistic logic. Its immediate corollary is a simple, constructive proof of the consistency of classical arithmetic from the assumption of the consistency of intuitionistic arithmetic.

The interpretation g of intuitionistic propositional logic into modal logic **S4** introduced in [126], where necessity $\Box$ could be interpreted as constructive provability, is defined as follows:

- $g(p)$ is $(\Box p)$, if p is atomic;
- $g(\neg\varphi)$ is $(\Box(\neg g(\varphi)))$;
- $g(\varphi \wedge \psi)$ is $(g(\varphi) \wedge g(\psi))$;
- $g(\varphi \vee \psi)$ is $(\Box(g(\varphi) \vee g(\psi)))$;
- $g(\varphi \Rightarrow \psi)$ is $(\Box(g(\varphi) \Rightarrow g(\psi)))$.

Another remarkable case is the well-known "correspondence theory" introduced in [264], which shows that propositional modal logic can be almost straightforwardly encoded in first-order classical logic. Indeed, using the so-called *standard translation*, a propositional modal logic can be viewed as a fragment of first-order classical logic. Moreover, any modal logic whose class of models has accessibility relations characterized by a first-order formula (for instance, **S4**) admits a weak and conservative translation into first-order classical logic.

In all the previous cases the collection of translations reduces to a singleton. All such examples are but instances of (weak) possible-translations characterizations.

Using Proposition 9.1.17, a representation of *PTC*s can be given in terms of products and conservative translations. The next result was originally stated in [30] for the category of compact and structural consequence systems.

Proposition 9.2.5 *Small grammatical possible-translations characterizations for a consequence system $\mathcal{C}$ are the same as conservative morphisms $f : \mathcal{C} \to \mathcal{C}'$, where $\mathcal{C}'$ is a product in* **sCon** *of some small family of consequence systems. Moreover, if $\mathcal{C}$ admits a small grammatical structural PTC then $\mathcal{C}$ is structural.*

Proof. Let $\mathcal{C} = \langle C, \vdash \rangle$ be a consequence system and let P be a small grammatical *PTC* for $\mathcal{C}$. The idea is to define a conservative morphism

$$\mathsf{t}(P) : \mathcal{C} \to \mathcal{C}(P)$$

in **sCon**, where $\mathcal{C}(P)$ is a product in **sCon** of some family of consequence systems, such that $\mathsf{t}(P)$ encodes P. Conversely, given a conservative morphism $f : \mathcal{C} \to \mathcal{C}'$ in **sCon**, where $\mathcal{C}'$ is a product in **sCon** of a small family of consequence systems, a small grammatical *PTC* for $\mathcal{C}$ encoding f, denoted $\mathsf{PTC}(f)$, can be defined, in such a manner that the mapping t is the inverse of PTC, and vice-versa.

Thus, assuming that $P = \langle \{\mathcal{C}_i\}_{i \in I}, \{f_i\}_{i \in I} \rangle$ is a small grammatical *PTC* for $\mathcal{C}$, consider the product $\langle \mathcal{C}^{\mathcal{F}}, \{\pi_i\}_{i \in I} \rangle$ in **sCon** of the small and non-empty family $\mathcal{F} = \{\mathcal{C}_i\}_{i \in I}$ (recall Proposition 9.1.16). Since each f_i is a morphism in **sCon** then, by the universal property of the product, there is a unique morphism

$$\mathsf{t}(P) : \mathcal{C} \to \mathcal{C}^{\mathcal{F}}$$

in **sCon** such that $f_i = \pi_i \centerdot \mathsf{t}(P)$ for every $i \in I$. By Lemma 9.1.5 it follows that

$$\widehat{f_i} = \widehat{\pi_i} \circ \widehat{\mathsf{t}(P)}\,.$$

Using this, it can be proved that $\mathsf{t}(P)$ is a conservative morphism. Clearly, $\mathsf{t}(P)$ together with its codomain $\mathcal{C}(P) = \mathcal{C}^{\mathcal{F}}$ encodes all the information about P: every consequence system $\mathcal{C}_i$ is obtained as the codomain of π_i, and every morphism f_i is obtained as $f_i = \pi_i \centerdot \mathsf{t}(P)$.

Conversely, let $f : \mathcal{C} \to \mathcal{C}'$ be a conservative morphism in **sCon**, such that $\mathcal{C}'$ is a product in **sCon** of a small and non-empty family $\{\mathcal{C}_i\}_{i \in I}$ of consequence systems, with canonical projections π_i for every $i \in I$. For every $i \in I$ consider the morphism $f_i = \pi_i \centerdot f$ in **sCon**, and define the small grammatical possible-translations frame

$$\mathsf{PTC}(f) = \langle \{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle.$$

Using Lemma 9.1.5 again, it can be proved that $\mathsf{PTC}(f)$ is a (small and grammatical) *PTC* for $\mathcal{C}$. Moreover, all the information about f and $\mathcal{C}'$ can be recovered from $\mathsf{PTC}(f)$: in fact

$$f = \mathsf{t}(\mathsf{PTC}(f))$$

and $\mathcal{C}'$ is the product of the family of consequence systems of $\mathsf{PTC}(f)$. It is also clear that, if P is a small grammatical *PTC* for $\mathcal{C}$, then $\mathsf{PTC}(\mathsf{t}(P)) = P$.

Finally, suppose that P is a small grammatical structural *PTC* for $\mathcal{C} = \langle C, \vdash\rangle$. Then, $\mathcal{C}(P) = \langle C', \vdash'\rangle$ is a structural consequence system, by Proposition 9.1.16. Suppose that $\Gamma \vdash \varphi$ and let σ be a substitution over C. Using Lemma 9.1.6, there exists a substitution σ' over C' such that

$$\widehat{\mathsf{t}(P)} \circ \widehat{\sigma} = \widehat{\sigma}' \circ \widehat{\mathsf{t}(P)}.$$

Since $\widehat{\mathsf{t}(P)}(\Gamma) \vdash' \widehat{\mathsf{t}(P)}(\varphi)$ then $\widehat{\sigma}'(\widehat{\mathsf{t}(P)}(\Gamma)) \vdash' \widehat{\sigma}'(\widehat{\mathsf{t}(P)}(\varphi))$, that is,

$$\widehat{\mathsf{t}(P)}(\widehat{\sigma}(\Gamma)) \vdash' \widehat{\mathsf{t}(P)}(\widehat{\sigma}(\varphi)).$$

It follows that $\widehat{\sigma}(\Gamma) \vdash \widehat{\sigma}(\varphi)$, because $\mathsf{t}(P)$ is conservative. This shows that $\mathcal{C}$ is structural. $\triangleleft$

With respect to compact consequence systems, the analogous to the proposition above is as follows.

Proposition 9.2.6 *Small grammatical and compact possible-translations characterizations for a consequence system $\mathcal{C}$ are the same as conservative morphisms $f : \mathcal{C} \to \mathcal{C}'$, where $\mathcal{C}'$ is a product in* **csCon** *of some small and non-empty family of compact consequence systems. As a consequence, if $\mathcal{C}$ admits a small grammatical and compact possible-translations characterization P then $\mathcal{C}$ is compact. If, additionally, P is structural then $\mathcal{C}$ is structural.*

Proof. The proof is analogous to that of Proposition 9.2.5, but now using Proposition 9.1.17. Details are left to the reader. $\triangleleft$

We now show that matrix semantics for propositional logics (referred to at the beginning of this chapter) can be portrayed as a particular case of *PTS*s (see Proposition 9.2.14 below). In order to do this, we briefly recall some basic facts about matrix semantics. Recall the definition of algebra over a signature given in Section 3.1 of Chapter 3.

Definition 9.2.7 Given a signature C, a *matrix* over C is a pair

$$M = \langle \mathcal{B}, D \rangle$$

where $\mathcal{B} = \langle B, \nu \rangle$ is an algebra over C and $D \subseteq B$. The set D is usually referred to as the *set of designated values of* M.

The homomorphisms $v : L(C) \to B$ over C are called *M-valuations* or *valuations over M*. ▽

Observe that every M-valuation $v : L(C) \to B$ is generated by a unique assignment $\alpha : \Xi \to B$. This fact will be frequently used from now on.

For simplicity, sometimes we will write $M = \langle B, D \rangle$ instead of $M = \langle \mathcal{B}, D \rangle$ in concrete examples. Additionally, the interpretation of a connective c in M will be frequently written as c^M. However, sometimes we will use the same symbol to refer both to a connective as well as its interpretation in a given matrix, following a common practice.

Definition 9.2.8 Let C be a signature and let $\mathcal{K}$ be a class of matrices over C. The *matrix semantics for $L(C)$ induced by $\mathcal{K}$*, denoted by $\models_{\mathcal{K}}$, is defined by:

$\Gamma \models_{\mathcal{K}} \varphi$ if and only if $v(\Gamma) \subseteq D$ implies that $v(\varphi) \in D$
for every matrix $M = \langle \mathcal{B}, D \rangle$ over C in $\mathcal{K}$
and every M-valuation v for $L(C)$.

▽

Observe that the notion of (global) entailment introduced in Definition 3.1.22 is a particular case of the notion of entailment presented above. In the former, the set of designated values is always of the form $\{\top\}$. The assignment α in Definition 3.1.22 corresponds to the restriction of a valuation v to the set Ξ in Definition 9.2.8.

A structural consequence system $\mathcal{C} = \langle C, \vdash \rangle$ is said to be a *matrix logic* if there exists a class $\mathcal{K}$ of matrices over C such that $\vdash \; = \; \vdash_{\mathcal{K}}$. In this case, we say that $\mathcal{K}$ is *adequate* for $\mathcal{C}$, or that $\mathcal{K}$ is a *complete matrix semantics* for $\mathcal{C}$, or that $\mathcal{C}$ is *characterized by* $\mathcal{K}$. As shown in [278], every structural consequence system is indeed a matrix logic (see Proposition 9.2.13 below). When $\mathcal{K} = \{M\}$ is a singleton then $\models_M$ will stand for $\models_{\{M\}}$. A matrix logic of this form was already considered in Example 1.4.6 of Chapter 1.

Example 9.2.9 The 3-valued weakly intuitionistic I^1 was introduced in [245], and studied in several contexts afterward.

The logic I^1 can be presented either axiomatically, or as a matrix logic over the signature C^{I^1} such that:

- $|C^{I^1}| = \{\neg, \Rightarrow\}$

Its matrix is

- $M_{I^1} = \langle \mathcal{B}_{I^1}, \{T\}\rangle$, where $B_{I^1} = \{T, F_1, F\}$.

The corresponding operations are displayed in the tables below.

	$\neg$
T	F
F_1	F
F	T

$\Rightarrow$	T	F_1	F
T	T	F	F
F_1	T	T	T
F	T	T	T

Observe that $\Rightarrow$ cannot distinguish the two non-designated values. Classical conjunction and disjunction can be defined in I^1 as follows:

$$\begin{aligned}(\varphi \wedge \psi) &= (\neg(((\varphi \Rightarrow \varphi) \Rightarrow \varphi) \Rightarrow (\neg((\psi \Rightarrow \psi) \Rightarrow \psi)))),\\ (\varphi \vee \psi) &= (((\neg(\psi \Rightarrow \psi)) \Rightarrow \psi) \Rightarrow ((\varphi \Rightarrow \varphi) \Rightarrow \varphi)).\end{aligned}$$

The tables for these connectives are showed below.

$\wedge$	T	F_1	F
T	T	F	F
F_1	F	F	F
F	F	F	F

$\vee$	T	F_1	F
T	T	T	T
F_1	T	F	F
F	T	F	F

The intuitionistic character of I^1 is evidenced by that fact that $(\xi \vee (\neg\,\xi))$ is not a theorem of I^1 for any schema variable ξ. On the other hand, $(\varphi \vee (\neg\varphi))$ is a theorem of I^1 for every complex (non-atomic) formula φ. The reader can easily check that, for instance,

$$\models_{M_{I^1}} (\varphi \Rightarrow (\neg(\neg\,\varphi)))$$

for every φ, and

$$\models_{M_{I^1}} ((\neg(\neg\,\varphi)) \Rightarrow \varphi)$$

for every non-atomic φ.

The logic I^1 is maximal with respect to classical logic in the following sense: adding to I^1 any classical tautology (which is not a theorem of I^1) as an axiom schema, produces a system which collapses with classical logic.

The maximality of I^1 with respect to classical logic can be characterized in lattice-theoretic terms. Let $CONS_{cs}(C)$ be the set of compact and structural consequence system defined over signature C. As observed above,

$$\mathcal{C} \leq \mathcal{C}' \text{ in } CONS_{cs}(C)$$

(where $\leq$ denotes the weakness relation introduced in Definition 1.1.6) if and only if the consequence relation of $\mathcal{C}$ is contained in the consequence relation of $\mathcal{C}'$. The set $CONS_{cs}(C)$, ordered by the weakness relation, is a complete lattice (see [280]). Let $\mathcal{C}_{I^1}$ and $\mathcal{C}$ be the consequence systems of I^1 and of classical logic defined over the signature C^{I^1}, respectively. Then $\mathcal{C}_{I^1} < \mathcal{C}$ (that is, $\mathcal{C}_{I^1}$ is properly contained in $\mathcal{C}$) and, if $\mathcal{C}' \in CONS_{cs}(C)$ is such that $\mathcal{C}_{I^1} < \mathcal{C}' \leq \mathcal{C}$ then $\mathcal{C}' = \mathcal{C}$. ∇

Definition 9.2.10 Let $\mathcal{C} = \langle C, \vdash \rangle$ be a structural consequence system, and let M be a matrix over C. We say that $\vdash$ is *sound for* $\models_M$, or that M *is a matrix model for* $\mathcal{C}$, if $\vdash \subseteq \models_M$. $\triangledown$

We define the class $\textsc{MatMod}(\mathcal{C})$ as being the class of all the matrix models for $\mathcal{C}$.

It is easy to prove that the pair $\langle C, \models_{\mathcal{K}} \rangle$ is a consequence system, for every class $\mathcal{K}$ of matrices over C, and $\langle C, \models_M \rangle$ is a structural consequence system, for every matrix M over C. Let $CONS_s(C)$ be the set of structural consequence systems defined over signature C, ordered with respect to the weakness relation, that is, with respect to inclusion of the respective consequence relations. Then $CONS_s(C)$ is a complete lattice (see [280]). The following fundamental result (stated in [183]) shows that a matrix logic is, in fact, structural:

Proposition 9.2.11 *Let $\mathcal{K}$ be a class of matrices over C. Then*

$$\langle C, \models_{\mathcal{K}} \rangle = \bigwedge \{ \langle C, \models_M \rangle \ : \ M \in \mathcal{K} \}$$

where the infimum $\bigwedge$ is taken in the complete lattice $CONS_s(C)$, and so $\langle C, \models_{\mathcal{K}} \rangle$ is a structural consequence system.

Note that $\langle C, \models_{\mathcal{K}} \rangle$ is not necessarily compact. However, in [279], a sufficient condition is obtained for a matrix logic to be standard.

Proposition 9.2.12 *Every structural consequence relation induced by a finite class of finite matrices is compact, and so defines a standard consequence system.*

The next result, due to Adolf Lindenbaum (see [278, 280]), establishes that every structural consequence system is indeed a matrix logic.

Proposition 9.2.13 *Let $\mathcal{C}$ be a structural consequence system. Then the class* $\textsc{MatMod}(\mathcal{C})$ *is a complete matrix semantics for $\mathcal{C}$.*

It is straightforward to check that the notion of matrix logics is a special case of grammatical structural possible-translations semantics:[1]

Proposition 9.2.14 *Let $\mathcal{C} = \langle C, \vdash \rangle$ be a matrix logic and let $\mathcal{K}$ be a class of matrices over C adequate for $\mathcal{C}$. For every $M \in \mathcal{K}$, let $\mathcal{C}_M = \langle C, \models_M \rangle$ and let $f_M : \mathcal{C} \to \mathcal{C}_M$ be the morphism in* **sCon** *induced by the identity morphism for C. Then the grammatical structural possible-translations frame*

$$\mathsf{PTS}(\mathcal{K}) = \langle \{\mathcal{C}_M\}_{M \in \mathcal{K}}, \{f_M\}_{M \in \mathcal{K}} \rangle$$

is a grammatical structural possible-translations semantics for $\mathcal{C}$.

[1] See [197] for other results about characterizations of general logics by means of possible-translations semantics.

Proof. Observe that f_M is, in fact, a morphism in **sCon**, since $\vdash \subseteq \models_M$. The proof of the proposition is immediate from Definition 9.2.3 and from the notion of adequate class of matrices. ◁

From the last proposition, it follows that any structural consequence system $\mathcal{C}$ characterized by a class of matrices $\mathcal{K}$ splits over the matrices in $\mathcal{K}$. In other words, every matrix in $\mathcal{K}$ can be seen as a factor of $\mathcal{C}$. Therefore, an adequate matrix semantics is a particular instance of the splitting method defined by possible-translations semantics. Clearly, if $\mathcal{K}$ is a proper class (instead of a set), then $\mathsf{PTS}(\mathcal{K})$ is not small.

The next example shows that a well-known characterization of propositional intuitionistic logic can be recast in terms of weak possible-translations semantics.

Example 9.2.15 Let $\mathcal{C}_{\mathsf{Int}} = \langle C_{\mathsf{Int}}, \vdash_{\mathsf{Int}}\rangle$ be the consequence system of the propositional intuitionistic logic. A classical result states that theoremhood in $\mathcal{C}_{\mathsf{Int}}$ is characterized by the class of matrices

$$\mathcal{K} = \{\langle \mathcal{H}, \{\top\}\rangle : \mathcal{H} \text{ is a Heyting algebra with top element } \top\}.$$

That is, φ is a theorem of $\mathcal{C}_{\mathsf{Int}}$ if and only if, for every Heyting algebra $\mathcal{H}$ and every valuation v over $\mathcal{H}$, $v(\varphi) = \top$. A proof of this result can be found, for instance, in [227].

Now, for every matrix $\mathrm{H} = \langle \mathcal{H}, \{\top\}\rangle$ in $\mathcal{K}$ consider the consequence system $\mathcal{C}_{\mathrm{H}} = \langle C_{\mathsf{Int}}, \models_{\mathrm{H}}\rangle$, and let f_{H} be the morphism $f_{\mathrm{H}} : \mathcal{C}_{\mathsf{Int}} \to \mathcal{C}_{\mathrm{H}}$ in **sCon** induced by the identity morphism for the signature C_{Int} of $\mathcal{C}_{\mathsf{Int}}$. Then

$$\mathsf{PTS}(\mathcal{K}) = \langle \{\mathcal{C}_{\mathrm{H}}\}_{\mathrm{H}\in\mathcal{K}}, \{f_{\mathrm{H}}\}_{\mathrm{H}\in\mathcal{K}}\rangle$$

is a grammatical weak possible-translations semantics for propositional intuitionistic logic $\mathcal{C}_{\mathsf{Int}}$. That is, $\vdash_{\mathsf{Int}} \varphi$ if and only if $\models_{\mathrm{H}} \varphi$ for every Heyting algebra $\mathcal{H}$. ▽

It is worth noting that matrix semantics are a particular case of global semantics (see Subsection 3.3.1 of Chapter 3), since they are based on non-ordered algebras. For instance, the matrix semantics for $\mathcal{C}_{\mathsf{Int}}$ considering above just represents global reasoning. That is, for every Heyting algebra $\mathcal{H}$ and every set $\Gamma \cup \{\varphi\}$ of formulas, $\Gamma \models_{\mathrm{H}} \varphi$ if and only if, for every homomorphism

$$v : L(C_{\mathsf{Int}}) \to H$$

$v(\Gamma) \subseteq \{\top\}$ implies $v(\varphi) = \top$. However, by using algebraic ordered structures it is frequently useful to consider the local notion of entailment, that is: φ is a consequence of Γ if and only if

$$(\bigwedge_{\gamma\in\Gamma} v(\gamma)) \leq v(\varphi)$$

for every homomorphism v (here, $\bigwedge$ denotes the infimum of a set). In the case of $\mathcal{C}_{\mathsf{Int}}$ analyzed in Example 9.2.15 nothing is lost: $\mathcal{C}_{\mathsf{Int}}$ is compact and satisfies the metatheorem of deduction, and so characterizing theoremhood is equivalent to characterizing local entailment.

Now we give an application of possible-translations semantics to paraconsistent logics. The next examples show that, in some cases, obtaining a PTC for a logic is more a necessity than a curiosity: certain logics are usually characterized by semantical methods which are not easy to manage. This is the case with most of paraconsistent logics defined in the literature. The PTSs we present below (as well as the PTS for **Ci** given in Chapter 1) produce relatively easy decision procedures for several **LFI**s which cannot be characterized by finite matrices (see [49]). This procedures can be used, as usual with semantic matters, to give counter-examples of entailment in the given logics.

Example 9.2.16 Consider again the *logics of formal inconsistency*, **LFI**s, briefly described in Example 5.1.10 of Chapter 5.

Some interesting **LFI**s are the logics **bC** and **Ci**, as well as its weaker versions **mCi** and **mbC**. The logic $\mathfrak{C}_1$ (recall Example 2.2.9 in Chapter 2) is also a well-known example of an **LFI**. In Chapter 1 we showed a PTS for **Ci** (recall Example 1.4.6). In this example we analyze the weaker logic **bC**. Like **Ci**, the logic **bC** is defined over a signature C such that

- $C_1 = \{\neg, \circ\}$;
- $C_2 = \{\wedge, \vee, \Rightarrow\}$;
- $C_k = \emptyset$ in any other case.

The Hilbert calculus for **bC** is presented below (see [51]).

- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1))\rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3)))\rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_1)\rangle$;
- $\langle \emptyset, ((\xi_1 \wedge \xi_2) \Rightarrow \xi_2)\rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow (\xi_1 \wedge \xi_2)))\rangle$;
- $\langle \emptyset, (\xi_1 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle \emptyset, (\xi_2 \Rightarrow (\xi_1 \vee \xi_2))\rangle$;
- $\langle \emptyset, ((\xi_1 \Rightarrow \xi_3) \Rightarrow ((\xi_2 \Rightarrow \xi_3) \Rightarrow ((\xi_1 \vee \xi_2) \Rightarrow \xi_3)))\rangle$;
- $\langle \emptyset, ((\neg(\neg \xi_1)) \Rightarrow \xi_1)\rangle$;
- $\langle \emptyset, (\xi_1 \vee (\neg \xi_1))\rangle$;

- $\langle \emptyset, ((\circ\xi_1) \Rightarrow (\xi_1 \Rightarrow ((\neg\,\xi_1) \Rightarrow \xi_2)))\rangle$;
- $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2\rangle$.

Notice that the axiomatization of **bC** is closely related to that of $\mathfrak{C}_1$. However, the reader should observe that, in the case of **bC**, the consistency of the formula φ is described by the formula $(\circ\varphi)$ instead of the formula $\varphi^\circ = (\neg(\varphi \wedge (\neg\,\varphi)))$ as in the case of $\mathfrak{C}_1$. Thus, **bC** is not a subsystem of $\mathfrak{C}_1$ as it could seem at first sight. As a matter of fact, the logic **Ci** is obtained from **bC** by adding the axiom schema

$$((\neg(\circ\xi_1)) \Rightarrow (\xi_1 \wedge (\neg\,\xi_1)))$$

(this fact will be used in the proof of Proposition 9.2.20).

At the semantical level, **bC** is characterized by the family of all the bivaluations $v : L(C) \to 2$ satisfying the properties (v1)-(v6) of **Ci**-bivaluations mentioned in Example 1.4.6 of Chapter 1 (as in [49]). Thus, a **bC**-valuation does not necessarily satisfy property (v7) of **Ci**-valuations.

Let $\mathcal{C}_{\mathbf{bC}} = \langle C, \models_{\mathbf{bC}}\rangle$ be the consequence system obtained by using the bivaluation semantics for **bC**. Now we will present an original small structural and compact PTS for $\mathcal{C}_{\mathbf{bC}}$.

Consider the signature C^1 such that

- $C^1_1 = \{\neg_1, \neg_2, \circ_1, \circ_2, \circ_3\}$;
- $C^1_2 = \{\wedge, \vee, \Rightarrow\}$;
- $C^1_k = \emptyset$ in any other case.

Let M be the matrix over C^1 with domain $\{T, t, F\}$ defined through the truth-tables below, where $\{T, t\}$ is the set of designated values.

$\wedge$	T	t	F
T	t	t	F
t	t	t	F
F	F	F	F

$\vee$	T	t	F
T	t	t	t
t	t	t	t
F	t	t	F

$\Rightarrow$	T	t	F
T	t	t	F
t	t	t	F
F	t	t	t

	$\neg_1$	$\neg_2$
T	F	F
t	F	t
F	T	T

	$\circ_1$	$\circ_2$	$\circ_3$
T	T	t	F
t	F	t	F
F	T	t	F

Let $\{f_i\}_{i\in I}$ be the family of all the mappings $f : L(C) \to L(C^1)$ satisfying clauses $(tr0)$, $(tr1)$, $(tr2)$, $(tr3)$ and $(tr4)$ below.

(tr0) $f(\xi) = \xi$ for $\xi \in \Xi$;

(tr1) $f(\neg\,\varphi) \in \{(\neg_1 f(\varphi)), (\neg_2 f(\varphi))\}$;

(tr2) $f(\varphi\#\psi) = (f(\varphi)\#f(\psi))$, for $\# \in \{\wedge, \vee, \Rightarrow\}$;

(tr3) $f(\circ\varphi) \in \{(\circ_1 f(\varphi)), (\circ_2 f(\varphi)), (\circ_3 f(\varphi))\}$;

(tr4) if $f(\neg\varphi) = (\neg_2 f(\varphi))$ then $f(\circ\varphi) = (\circ_1 f(\varphi))$.

It will be proved below (see Proposition 9.2.19) that the family of mappings $\{f_i\}_{i\in I}$ defines a PTS for the logic **bC**. $\triangledown$

Proposition 9.2.17 *If a mapping $f : L(C) \to L(C^1)$ satisfies the clauses $(tr0)$ – $(tr4)$ above then f is a translation between $\mathcal{C}_{\mathbf{bC}}$ and the matrix logic $\langle C^1, \models_M\rangle$. Thus, the pair*

$$\mathsf{PTS} = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$$

is a small structural and compact possible-translations frame for the consequence system $\mathcal{C}_{\mathbf{bC}}$, where $\mathcal{C}_i = \langle C^1, \models_M\rangle$ for every $i \in I$.

Proof. It is sufficient to verify that the (finite) collection of all possible translations of each axiom produces tautologies in the matrix logic $\langle C^1, \models_M\rangle$, and that all possible translations of the rule of modus ponens preserve validity in $\langle C^1, \models_M\rangle$. Details of the proof are left to the reader. $\triangleleft$

In order to prove that PTS is, in fact, a possible-translations semantics for $\mathcal{C}_{\mathbf{bC}}$, we state the following lemma.

Lemma 9.2.18 *Given a* **bC***-valuation v there is a translation f in PTS and a M-valuation w such that, for every $\varphi \in L(C)$,*

$$v(\varphi) = 1 \text{ if and only if } w(f(\varphi)) \in \{T, t\}.$$

Proof. Consider the M-valuation $w : L(C^1) \to \{T, t, F\}$ such that

$$w(\xi) = \begin{cases} T & \text{if } v(\neg\xi) = 0 \\ t & \text{if } v(\xi) = v(\neg\xi) \\ F & \text{if } v(\xi) = 0 \end{cases}$$

for every $\xi \in \Xi$. Observe that, if $v(\varphi) = v(\neg\varphi)$ then $v(\varphi) = v(\neg\varphi) = 1$. Now, define recursively a mapping $f : L(C) \to L(C^1)$ as follows.

- $f(\xi) = \xi$ for $\xi \in \Xi$
- $f(\neg\varphi) = \begin{cases} (\neg_2 f(\varphi)) & \text{if } v(\varphi) = v(\neg\varphi) \\ (\neg_1 f(\varphi)) & \text{otherwise} \end{cases}$

- $f(\circ\varphi) = \begin{cases} (\circ_1 f(\varphi)) & \text{if } v(\varphi) = v(\neg\,\varphi) \\ (\circ_2 f(\varphi)) & \text{if } v(\varphi) \neq v(\neg\,\varphi) \text{ and } v(\circ\varphi) = 1 \\ (\circ_3 f(\varphi)) & \text{if } v(\varphi) \neq v(\neg\,\varphi) \text{ and } v(\circ\varphi) = 0 \end{cases}$
- $f(\varphi\#\psi) = (f(\varphi)\#f(\psi))$ for $\# \in \{\wedge, \vee, \Rightarrow\}$.

Clearly, the mapping f satisfies the clauses $(tr0) - (tr4)$ above. In fact, the unique clause that deserves a close analysis is clause $(tr4)$. Thus, suppose that $f(\neg\,\varphi) = (\neg_2\, f(\varphi))$. Then $v(\varphi) = v(\neg\,\varphi)$, by construction of f, and therefore $f(\circ\varphi) = (\circ_1 f(\varphi))$, using again the definition of f. This shows that t satisfies $(tr4)$.

In order to prove that

$$v(\varphi) = 1 \;\text{ if and only if }\; w(f(\varphi)) \in \{T, t\}$$

for every $\varphi \in L(C)$, we need to define the notion of complexity $l(\varphi)$ of a formula φ in $L(C)$. This mapping is defined as follows:

- $l(\xi) = 1$ if $\xi \in \Xi$;
- $l(\neg\,\varphi) = l(\varphi) + 1$;
- $l(\circ\varphi) = l(\varphi) + 2$;
- $l(\varphi\#\psi) = l(\varphi) + l(\psi) + 1$ for $\# \in \{\wedge, \vee, \Rightarrow\}$.

Now we prove by induction on $l(\varphi)$ the following: for every $\varphi \in L(C)$,

(1) $v(\varphi) = 1$ if and only if $w(f(\varphi)) \in \{T, t\}$;
(2) if $v(\varphi) = v(\neg\,\varphi)$ then $w(f(\varphi)) = w(f(\neg\,\varphi)) = t$.

The case $\varphi \in \Xi$ is straightforward. Assume that (1) and (2) hold, for every φ with $l(\varphi) \leq n$ (for $n \geq 1$) and let $\varphi \in L(C)$ with $l(\varphi) > n$. There are three cases to analyze.

Case 1: φ is $(\neg\,\psi)$. We first prove that (1) holds in this case.

Suppose that $v(\varphi) = 1$, that is, $v(\neg\,\psi) = 1$. If $v(\psi) = 1$ then, by induction hypothesis (2), $w(f(\psi)) = w(f(\neg\,\psi)) = t$ and so $w(f(\varphi)) \in \{T, t\}$. If $v(\psi) = 0$ then, by induction hypothesis (1),

$$w(f(\psi)) = F$$

and so

$$w(f(\neg\,\psi)) = T$$

by construction of f and by the truth-tables of M. That is, $w(f(\varphi)) \in \{T, t\}$. Conversely, suppose that $w(f(\varphi)) \in \{T, t\}$. If $f(\neg\,\psi) = (\neg_2\, f(\psi))$ then

$$v(\psi) = v(\neg\,\psi) = 1$$

by construction of f, and so $v(\varphi) = 1$. On the other hand, if $f(\neg\psi) = (\neg_1 f(\psi))$ then

$$w(f(\psi)) = F$$

by the truth-table for $\neg_1$, and so $v(\psi) = 0$, by induction hypothesis (1). Then $v(\neg\psi) = 1$, by the property (v4) of **bC**-bivaluations (see Example 1.4.6); that is, $v(\varphi) = 1$ and so (1) holds.

In order to prove (2), assume that $v(\neg\psi) = v(\neg(\neg\psi)) = 1$. Then

$$f(\neg(\neg\psi)) = (\neg_2 f(\neg\psi))$$

by construction of f. On the other hand, $v(\neg(\neg\psi)) = 1$ implies $v(\psi) = 1$, by the property (v5) of v (recall Example 1.4.6), and so $v(\psi) = v(\neg\psi)$. Then

$$w(f(\psi)) = w(f(\neg\psi)) = t$$

by induction hypothesis (2), therefore

$$w(f(\neg(\neg\psi))) = w(\neg_2 f(\neg\psi)) = \neg_2 w(f(\neg\psi)) = (\neg_2 t) = t.$$

Thus $w(f(\varphi)) = w(f(\neg\varphi)) = t$.

Case 2: φ is $(\circ\psi)$. We first prove the property (1).

Thus, suppose that $v(\varphi) = 1$, that is, $v(\circ\psi) = 1$. By the property (v6) of v (recall Example 1.4.6) it follows that $v(\psi) = 0$ or $v(\neg\psi) = 0$. That is,

$$v(\psi) \neq v(\neg\psi) \text{ and } v(\circ\psi) = 1$$

thus $f(\circ\psi) = (\circ_2 f(\psi))$, by construction of f. Then

$$w(f(\varphi)) = w(f(\circ\psi)) = (\circ_2 w(f(\psi))) = t$$

by the truth-table for $\circ_2$, and so $w(f(\varphi)) \in \{T, t\}$. Conversely, assume that $w(f(\varphi)) \in \{T, t\}$. If $v(\psi) = v(\neg\psi)$ then $w(f(\psi)) = w(f(\neg\psi)) = t$, by induction hypothesis (2). Using the definition of f we infer that $f(\circ\psi) = (\circ_1 f(\psi))$ and so, from the truth-tables for $\circ_1$ it follows that

$$w(f(\circ\psi)) = (\circ_1 w(f(\psi))) = (\circ_1 t) = F$$

a contradiction. Therefore $v(\psi) \neq v(\neg\psi)$. Now, if $v(\circ\psi) = 0$ then, by construction of f, $f(\circ\psi) = (\circ_3 f(\psi))$ and so, from the truth-tables for $\circ_3$ it follows that $w(f(\circ\psi)) = (\circ_3 w(f(\psi))) = F$, a contradiction. Thus $v(\circ\psi) = 1$, that is, $v(\varphi) = 1$. Therefore (1) holds in this case.

In order to prove (2), suppose that $v(\circ\psi) = v(\neg(\circ\psi))$. Then $v(\circ\psi) = 1$ and so we infer that $w(f(\varphi)) = t$, as it was proved at the beginning of Case 2. On the other hand, from $v(\circ\psi) = v(\neg(\circ\psi))$ it follows that

$$f(\neg(\circ\psi)) = (\neg_2 f(\circ\psi))$$

by definition of f. Then

$$w(f(\neg\,\varphi)) = w(f(\neg(\circ\psi))) = (\neg_2 w(f(\circ\psi))) = (\neg_2 t) = t.$$

This proves (2) for this case.

Case 3: φ is $(\psi\#\delta)$ for $\# \in \{\wedge, \vee, \Rightarrow\}$. Observe that the truth-tables for these operators enjoy the following property:

- $w(f(\psi \wedge \delta)) \in \{T, t\}$ if and only if $w(f(\psi)) \in \{T, t\}$ and $w(f(\delta)) \in \{T, t\}$;
- $w(f(\psi \vee \delta)) \in \{T, t\}$ if and only if $w(f(\psi)) \in \{T, t\}$ or $w(f(\delta)) \in \{T, t\}$;
- $w(f(\psi \Rightarrow \delta)) \in \{T, t\}$ if and only if $w(f(\psi)) = F$ or $w(f(\delta)) \in \{T, t\}$.

From this and from the properties (v1)-(v3) of v (recall Example 1.4.6) and by induction hypothesis (1), the property (1) hold in this case. In order to prove (2), assume that $v(\psi\#\delta) = v(\neg(\psi\#\delta))$. Then $v(\psi\#\delta) = 1$ and so

$$w(f(\psi\#\delta)) \in \{T, t\}$$

since we already prove (1) in this case. But then $w(f(\psi\#\delta)) = t$, by the definition of f and by the truth-tables of M. On the other hand, by definition of f it follows that $f(\neg(\psi\#\delta)) = \neg_2\, f(\psi\#\delta)$ and then

$$w(f(\neg(\psi\#\delta))) = (\neg_2 w(f(\psi\#\delta))) = (\neg_2 t) = t.$$

Therefore (2) holds in this case.

This completes the proof of the lemma. ◁

Proposition 9.2.19 *The small structural and compact possible-translations frame*

$$\mathsf{PTS} = \langle \{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$$

of Proposition 9.2.17 is a small structural and compact possible-translations semantics for the consequence system $\mathcal{C}_{\mathbf{bC}}$.

Proof. Let $\Gamma\cup\{\varphi\} \subseteq L(C)$. If $\Gamma \models_{\mathbf{bC}} \varphi$ then $f_i(\Gamma) \models_M f_i(\varphi)$ for every $i \in I$, since every mapping f_i is a translation. Conversely, suppose that $f_i(\Gamma) \models_M f_i(\varphi)$ for every $i \in I$. Let v be a **bC**-valuation such that $v(\gamma) = 1$ for every $\gamma \in \Gamma$. Consider a translation f in PTS and a M-valuation w such that, for every $\psi \in L(C)$,

$$v(\psi) = 1 \text{ if and only if } w(f(\psi)) \in \{T, t\}.$$

The existence of such w and f is guaranteed by Lemma 9.2.18. Then $w(f(\gamma)) \in \{T, t\}$ for every $\gamma \in \Gamma$ and so $w(f(\varphi)) \in \{T, t\}$, because $f(\Gamma) \models_M f(\varphi)$. Therefore $v(\varphi) = 1$ and so $\Gamma \models_{\mathbf{bC}} \varphi$. ◁

Now we are ready to prove that the PTS proposed for **Ci** in Example 1.4.6 works fine.

Proposition 9.2.20 *The pair*

$$P = \langle \{\mathcal{C}_f\}_{f \in Tr}, Tr \rangle$$

defined in Example 1.4.6 of Chapter 1 is a small structural and compact possible-translations semantics for the consequence system $\mathcal{C}_{\mathbf{Ci}}$ *of the logic* **Ci**.

Proof. The proof is similar to that for **bC**. Recall that $\mathcal{C}_f$ is the matrix logic $\mathcal{C}$ characterized by the matrix M_0 given in Example 1.4.6, for every $f \in Tr$. The fact that every mapping f in Tr is a translation between $\mathcal{C}_{\mathbf{Ci}}$ and $\mathcal{C}_f$ is easily proved by checking that every translation of every axiom of **Ci** is validated by the matrix M_0, and by observing that validity in M_0 is preserved by modus ponens. Then P is a possible-translations frame for $\mathcal{C}_{\mathbf{Ci}}$.

In order to prove that the frame P is a PTS for $\mathcal{C}_{\mathbf{Ci}}$, a result analogous to Lemma 9.2.18 can be obtained for P. The proof of the lemma is similar to the given above for Lemma 9.2.18, with the following changes: the mappings w and f are defined as above, but in this case

$$f(\circ\varphi) = \begin{cases} (\circ_1 f(\varphi)) & \text{if } v(\varphi) = v(\neg\varphi) \\ \\ (\circ_2 f(\varphi)) & \text{otherwise} \end{cases}$$

(recall that now the truth-table for $\circ_2$ is the one given in Example 1.4.6). By induction on the complexity of φ (defined as in the proof of Lemma 9.2.18) it can be proved that

(1) $v(\varphi) = 1$ if and only if $w(f(\varphi)) \in \{T, t\}$;

(2) if $v(\varphi) = v(\neg\varphi)$ then $w(f(\varphi)) = w(f(\neg\varphi)) = t$.

The proof is identical to that of Lemma 9.2.18, with the exception of the case when φ is $\circ\psi$.

Thus, suppose that $v(\circ\psi) = 1$. Then either $v(\psi) = 0$ or $v(\neg\psi) = 0$ and so $v(\psi) \neq v(\neg\psi)$. Thus $f(\circ\psi) = (\circ_2 f(\psi))$, therefore

$$w(f(\circ\psi)) = (\circ_2 w(f(\psi))) = T.$$

Conversely, suppose that $w(f(\circ\psi)) \in \{T, t\}$. If $v(\psi) = v(\neg\psi)$ then

$$w(f(\psi)) = w(f(\neg\psi)) = t$$

by induction hypothesis (2), and $f(\circ\psi) = (\circ_1 f(\psi))$, by the definition of f. Therefore

$$w(f(\circ\psi)) = (\circ_1 w(f(\psi))) = (\circ_1 t) = F$$

a contradiction. Thus $(v(\psi) \neq v(\neg\psi))$ and so, by properties (v7) and (v4) of v (see Example 1.4.6), $v(\circ\psi) = 1$. This proves (1) for this case.

Now, suppose that $v(\circ\psi) = v(\neg(\circ\psi))$. Then $v(\circ\psi) = 1$ and so either $v(\varphi) = 0$ or $v(\neg\varphi) = 0$, by property (v6) of v. Since $v(\neg(\circ\varphi)) = 1$ then, by property (v7) of

v, $v(\varphi) = 1$ and $v(\neg\varphi) = 1$, a contradiction. Therefore it is not possible to have $v(\circ\psi) = v(\neg(\circ\psi))$ and then property (2) holds trivially in this case.

Since the analogous of Lemma 9.2.18 holds for P, the rest of the proof is identical to the proof of Proposition 9.2.19. $\triangleleft$

Next example introduces a PTS for the paraconsistent logic $\mathfrak{C}_1^{\neg\neg}$.

Example 9.2.21 In 1963 (see [73], see also [72]) da Costa proposed a whole hierarchy of paraconsistent propositional calculi, the logics $\mathfrak{C}_n$, for $0 < n < \omega$, already discussed in this book. One of his guidelines for proposing such calculi were that the new logics "should contain the most part of the schemata and rules of the classical propositional calculus which do not interfere with the first conditions".

The systems $\mathfrak{C}_n$ as proposed, however, did not fully accomplish this requirement: $\mathfrak{C}_1$, for instance, can be augmented with the classically valid principles $(\xi \Rightarrow (\neg(\neg\xi)))$ and $((\neg((\neg\xi) \wedge \xi)) \Rightarrow (\neg(\xi \wedge (\neg\xi))))$ without colliding with classical logic.

An extension of the logics $\mathfrak{C}_n$ with such principles, already suggested in the folklore of paraconsistency, was studied for the first time in [46], and possible-translations semantics for these logics, called $\mathfrak{C}_n^{\neg\neg}$, were provided.

It should be noted that there exists an asymmetry in the logics $\mathfrak{C}_n$ with respect to the consistency operator. For instance, in $\mathfrak{C}_1$ the consistency of a formula φ is represented by the formula $\varphi^\circ = (\neg(\varphi \wedge (\neg\varphi)))$. However, it could also be represented by $\varphi^\triangleright = (\neg((\neg\varphi) \wedge \varphi))$. It happens that $\varphi^\triangleright$ follows from φ° in $\mathfrak{C}_1$, but the converse is not true. This motivates the inclusion of the axiom schema $(\xi^\triangleright \Rightarrow \xi^\circ)$ to $\mathfrak{C}_1$ (together with the axiom schema $(\xi \Rightarrow (\neg(\neg\xi)))$ guaranteeing the equivalence between φ and $(\neg(\neg\varphi))$ for every φ) in order to obtain $\mathfrak{C}_1^{\neg\neg}$.

We exhibit here a possible-translations semantics for $\mathfrak{C}_1^{\neg\neg}$, the first member of the hierarchy, taken from [46].

Let C^2 be the following signature:

- $C_1^2 = \{\neg_1, \neg_2\}$;
- $C_2^2 = \{\wedge_1, \wedge_2, \wedge_3, \vee_1, \vee_2, \vee_3, \Rightarrow_1, \Rightarrow_2, \Rightarrow_3\}$;
- $C_k^2 = \emptyset$ in any other case.

Let M' be the matrix over C^2 with domain $\{T, t, F\}$ defined by the following truth-tables, where $\{T, t\}$ is the set of designated values.

$\wedge_1$	T	t	F
T	T	T	F
t	t	t	F
F	F	F	F

$\wedge_2$	T	t	F
T	T	t	F
t	T	t	F
F	F	F	F

$\wedge_3$	T	t	F
T	T	t	F
t	t	t	F
F	F	F	F

$\vee_1$	T	t	F
T	T	T	T
t	t	t	t
F	T	T	F

$\vee_2$	T	t	F
T	T	t	T
t	T	t	T
F	T	t	F

$\vee_3$	T	t	F
T	T	t	T
t	t	t	t
F	T	t	F

$\Rightarrow_1$	T	t	F
T	T	T	F
t	t	t	F
F	T	T	T

$\Rightarrow_2$	T	t	F
T	T	t	F
t	T	t	F
F	T	t	T

$\Rightarrow_3$	T	t	F
T	T	t	F
t	t	t	F
F	T	t	T

	$\neg_1$
T	F
t	F
F	T

	$\neg_2$
T	F
t	t
F	T

Let $\{f_i\}_{i\in I}$ be the family of all the mappings $f_i : L(C) \to L(C^1)$ satisfying clauses $(tr0) - (tr8)$ below.

(tr0) $f(\xi) = \xi$ and $f(\neg\xi) = (\neg_2\xi)$, for $\xi \in \Xi$;

(tr1) $f(\varphi\#\psi) = (f(\varphi)\#_1 f(\psi))$ if $f(\neg\varphi) = (\neg_2 f(\varphi))$ and $f(\neg\psi) = (\neg_1 f(\psi))$, for $\# \in \{\wedge, \vee, \Rightarrow\}$;

(tr2) $f(\varphi\#\psi) = (f(\varphi)\#_2 f(\psi))$ if $f(\neg\varphi) = (\neg_1 f(\varphi))$ and $f(\neg\psi) = (\neg_2 f(\psi))$, for $\# \in \{\wedge, \vee, \Rightarrow\}$;

(tr3) $f(\varphi\#\psi) = (f(\varphi)\#_3 f(\psi))$ otherwise, for $\# \in \{\wedge, \vee, \Rightarrow\}$;

(tr4) $f(\neg(\varphi \wedge (\neg\varphi))) = (\neg_1\, f(\varphi \wedge (\neg\varphi)))$;

(tr5) $f(\neg((\neg\varphi) \wedge \varphi)) = (\neg_1\, f((\neg\varphi) \wedge \varphi))$;

(tr6) $f(\neg(\varphi\#\psi)) = (\neg_1 f(\varphi\#\psi))$ if $f(\neg\varphi) = (\neg_1 f(\varphi))$ and $f(\neg\psi) = (\neg_1 f(\psi))$, for $\# \in \{\wedge, \vee, \Rightarrow\}$ and $(\varphi\#\psi) \notin \{(\gamma \wedge (\neg\gamma)), ((\neg\gamma) \wedge \gamma)\}$ for every γ;

(tr7) $f(\neg(\varphi\#\psi)) \in \{(\neg_1 f(\varphi\#\psi)), (\neg_2 f(\varphi\#\psi))\}$ otherwise, for $\# \in \{\wedge, \vee, \Rightarrow\}$ and $(\varphi\#\psi) \notin \{(\gamma \wedge (\neg\gamma)), ((\neg\gamma) \wedge \gamma)\}$ for every γ;

(tr8) $f(\neg(\neg\varphi)) = (\neg_2 f(\neg\varphi))$.

Let $P = \langle\{\mathcal{C}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$, where $\mathcal{C}_i = \langle C^2, \models_{M'}\rangle$ is the consequence system associated to the matrix M', for every $i \in I$. Then P is a PTS for $\mathfrak{C}_1^{\neg\neg}$ (see [46]). $\triangledown$

The examples above show that grammatical PTSs do not cover many cases, and several logics as the **LFI**s above mentioned could require a PTS which is not grammatical. Of course, the fact that no grammatical PTSs are known for the logics in the examples above does not imply the non-existence of such characterizations.

9.3 Plain fibring of matrices

In this section the method of plain fibring for combining matrix logics, proposed in [60] and [94], is discussed. This technique, together with the simpler case, direct union of matrices, can be seen as an instance of fibring by functions (recall Chapter 1).

Briefly, the plain fibring takes two matrix logics $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C}_i$ is characterized by a single matrix M_i with domain B_i, and extends the original operators of the algebra M_i to the disjoint union $B_1 \oplus B_2$ by means of mappings $f_i : B_j \to B_i$ (for $i \neq j$).

The functions f_i 'transport' the truth-values of the matrix M_j into the truth-values of M_i, playing a role similar to the mappings from worlds into Kripke structures of the so-called fibring by functions, described in Subsection 1.2.3 of Chapter 1.

Plain fibring was designed as a method for splicing matrix logics into a new one, in a way that resembles the original proposal of fibring. However, this method can be see as a splitting method: in fact, as it was shown in Section 9.2, matrix logics are particular cases of possible-translation semantics, in which each matrix is a factor of the logic being represented. Moreover, a logic characterized by a single matrix can be considered as splitting through the logics defined by its sub-signatures.

A few words about notation: since the operations between matrices logics to be defined below are defined in terms of the matrices instead of the consequence relations, in the rest of this section we will write $\langle C, M \rangle$ instead of $\langle C, \models_M \rangle$ (note that two different matrices over C can define the same consequence relation).

It is convenient to begin with the simpler case, direct union of matrices. Given two matrix logics in which the respective algebras have the same domain and the same sets of designated values, then their direct union is a new matrix logic obtained simply by putting together both matrices. In formal terms:

Definition 9.3.1 Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (with $i = 1, 2$) be two matrix logics, where each $M_i = \langle \mathcal{B}_i, D_i \rangle$ is a matrix over C^i such that $B_1 = B_2$ and $D_1 = D_2$. Let $B = B_1$ and $D = D_1$. The *direct union of* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ is the consequence system

$$\mathcal{C}_1 + \mathcal{C}_2 = \langle C^1 \oplus C^2, \models_{M_1 + M_2} \rangle$$

such that:

- $C^1 \oplus C^2$ is the disjoint union of C^1 and C^2;
- $\models_{M_1+M_2}$ is the consequence relation induced by the matrix over $C^1 \oplus C^2$,
$$M_1 + M_2 = \langle \mathcal{B}, D \rangle;$$
- the matrix $M_1 + M_2$, with domain B, is defined as follows: if $c \in C^i_k$ and $b_1, \ldots, b_k \in B$, $k \geq 0$, then
$$c^{M_1+M_2}(b_1, \ldots, b_k) = c^{M_i}(b_1, \ldots, b_k)$$
for $i = 1, 2$. $\triangledown$

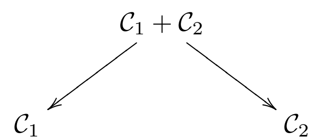
In the definition above, it is worth noting that the condition $B_1 = B_2$ and $D_1 = D_2$ does not mean that the operations defined in M_1 and M_2 coincide, that is: the interpretation mappings ν_1 and ν_2 of M_1 and M_2 do not necessarily coincide.

The fact that the direct union of matrices is a splitting operation is explained by the following result.

Proposition 9.3.2 *Let $\mathcal{C} = \langle C, \vdash \rangle$ be a consequence system characterized by a matrix M over C. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two fragments of $\mathcal{C}$ over C^1 and C^2, respectively, such that $C^1 \oplus C^2 = C$. Then $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$.*

Proof. Immediate from the definitions. ◁

The proposition above clarifies at what extent the direct union of consequence systems splits a consequence systems into simpler ones: any partition of the signature of a matrix logic into sub-signatures produces simpler matrix logics into which the original logic splits. In particular, a given consequence system $\mathcal{C}$ can be split into two simpler factors $\mathcal{C}_1$ and $\mathcal{C}_2$ whenever $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$ such that $\mathcal{C}$ is a conservative extension of both factors, by Proposition 9.1.14.



Of course this is a extremely simple way to decompose a logic into simpler components but, as we shall see in some examples below, this decomposition, together with the technique of plain fibring, can throw some light for understanding how logic systems are constructed.

On the other hand, direct union is also a splicing method, by the very definition, synthesizing the logics $\mathcal{C}_1$ and $\mathcal{C}_2$ into the complex logic $\mathcal{C}_1 + \mathcal{C}_2$. The new consequence system is a conservative extension of the given systems, showing that the conservativeness of $\mathcal{C}_1 + \mathcal{C}_2$ with respect of $\mathcal{C}_1$ and $\mathcal{C}_2$ is *created* (when splicing) and *derived* (when splitting).

Proposition 9.3.3 *Suppose that $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$. Then the system $\mathcal{C}$ is a conservative extension of both $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Proof. Straightforward from the definitions. ◁

Example 9.3.4 Recall the 3-valued Gödel logic from Examples 2.2.7 (Chapter 2) and 3.1.10 (Chapter 3). It is immediate to see from these examples that this logic can be presented by a consequence system $\mathcal{C} = \langle C, \models_M \rangle$ such that C is the intuitionistic signature of Example 2.1.4, that is: $C_1 = \{\neg\}$, $C_2 = \{\Rightarrow, \wedge, \vee\}$, and $C_k = \emptyset$ in any other case. By its turn, M is the matrix over C with domain

$\{\bot, 1/2, \top\}$, where $\top$ is the unique designated value, and the operations in M are defined by the truth-tables below.

	$\neg$
$\top$	$\bot$
$1/2$	$\bot$
$\bot$	$\top$

$\Rightarrow$	$\top$	$1/2$	$\bot$
$\top$	$\top$	$1/2$	$\bot$
$1/2$	$\top$	$\top$	$\bot$
$\bot$	$\top$	$\top$	$\top$

$\wedge$	$\top$	$1/2$	$\bot$
$\top$	$\top$	$1/2$	$\bot$
$1/2$	$1/2$	$1/2$	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\vee$	$\top$	$1/2$	$\bot$
$\top$	$\top$	$\top$	$\top$
$1/2$	$\top$	$1/2$	$1/2$
$\bot$	$\top$	$1/2$	$\bot$

Consider the sub-signatures C^1 and C^2 of C such that $C^1_1 = \{\neg\}$, $C^1_2 = \{\Rightarrow\}$, $C^2_2 = \{\wedge, \vee\}$ and $C^i_k = \emptyset$ in any other case. Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ such that M_i is the matrix over C^i corresponding to the truth-tables above (where $\top$ is again the only designated value), for $i = 1, 2$. That is, $\mathcal{C}_1$ is the fragment of 3-valued Gödel logic corresponding to negation and implication, while $\mathcal{C}_2$ is the fragment corresponding to conjunction and disjunction. Then

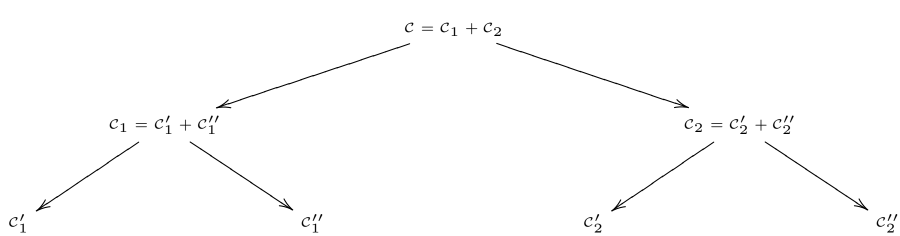
$$\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$$

and so the consequence systems $\mathcal{C}_1$ and $\mathcal{C}_2$ are two (simpler) factors of $\mathcal{C}$. On the other hand, $\mathcal{C}_1$ can split into two elementary consequence systems $\mathcal{C}'_1$ (the fragment of negation) and $\mathcal{C}''_1$ (the fragment of implication), that is,

$$\mathcal{C}_1 = \mathcal{C}'_1 + \mathcal{C}''_1.$$

By its turn, $\mathcal{C}_2$ splits into $\mathcal{C}'_2$ (the fragment of conjunction) and $\mathcal{C}''_2$ (the fragment of disjunction), that is, $\mathcal{C}_2 = \mathcal{C}'_2 + \mathcal{C}''_2$. Therefore the 3-valued Gödel logic can be factored as shown in the picture below. ∇

In [59] it was analyzed the question of recovering, by means of fibring, a logic from its fragments (recall Definition 9.1.11), showing that this question is closely



related to the definition of translation between logics to be considered. Thus, it was shown that, by using algebraic fibring (as defined in the previous chapters), a logic system cannot be obtained from its fragments, unless a stronger notion of translation preserving metaproperties be considered.

Direct union of matrices is just a part of a more elaborate process to be described now, in which two matrix logics defined over different domains are to be combined. The idea of the method, related to fibring by functions, is to extended each given matrix to the disjoint union of the domains by means of a pair of mappings, and then take the direct union of the extended matrices. The set of matrices obtained by taking all the possible pair of mappings (satisfying certain natural restrictions) defines a matrix semantics called *plain fibring*, to be introduced in Definition 9.3.7 below.

Definition 9.3.5 For $i = 1, 2$ consider a matrix logic $\mathcal{C}_i = \langle C^i, M_i \rangle$ defined by a single matrix $M_i = \langle \mathcal{B}_i, D_i \rangle$ with domain B_i over C^i.

- A pair of mappings (f_1, f_2) in $B_1^{B_2} \times B_2^{B_1}$ is *admissible* if it satisfies:

$$f_i(x) \in D_i \ \text{ if and only if } \ x \in D_j$$

  for every $x \in B_j$ (with $i \neq j$).

- Let $\mathsf{a} = (f_1, f_2)$ in $B_1^{B_2} \times B_2^{B_1}$ and $i \in \{1, 2\}$. The *extension of* M_i *by* a is the matrix over C^i,

$$M_i^{\mathsf{a}} = \langle \mathcal{B}, D_1 \oplus D_2 \rangle$$

  such that $B = B_1 \oplus B_2$ and, for every $c \in C^i_n$ and every $x_1, \ldots, x_n \in B$,

$$c^{M_i^{\mathsf{a}}}(x_1, \ldots, x_n) = c^{M_i}(\tilde{x}_1, \ldots, \tilde{x}_n)$$

  where, for every $k = 1, \ldots, n$:

$$\tilde{x}_k = \begin{cases} x_k & \text{if } x_k \in B_i \\ f_i(x_k) & \text{if } x_k \in B_j \ (\text{for } j \neq i) \end{cases}$$

∇

It is worth noting that, whenever a is admissible, the matrix logic $\mathcal{C}_i^{\mathsf{a}} = \langle C^i, M_i^{\mathsf{a}} \rangle$ obtained by extending M_1 to $B_1 \oplus B_2$ as above coincides with $\mathcal{C}_i$.

Proposition 9.3.6 *Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (with $i = 1, 2$) be two matrix logics, and let a be an admissible pair. Then $\Gamma \models_{M_i} \varphi$ if and only if $\Gamma \models_{M_i^{\mathsf{a}}} \varphi$, for every $\Gamma \cup \{\varphi\} \subseteq L(C^i)$ and $i = 1, 2$.*

Proof. We only prove the case for $i = 1$, because the other case is analogous.

Thus, let $\Gamma \cup \{\varphi\} \subseteq L(C^1)$. Suppose that $\Gamma \models_{M_1^{\mathsf{a}}} \varphi$, and let v be a valuation over M_1 such that $v(\Gamma) \subseteq D_1$. Then v can be considered as a valuation over M_1^{a} such that $v(\Gamma) \subseteq D_1 \oplus D_2$ and so $v(\varphi) \in D_1 \oplus D_2$. Thus $v(\varphi) \in D_1$ (because the image of v is contained in B_1) and so $\Gamma \models_{M_1} \varphi$.

Conversely, suppose that $\Gamma \models_{M_1} \varphi$ and let v' be a valuation over M_1^{a} such that $v'(\Gamma) \subseteq D_1 \oplus D_2$. Consider the valuation v over M_1 such that, for every $\xi \in \Xi$,

$$v(\xi) = \begin{cases} f_1(v'(\xi)) & \text{if } v'(\xi) \in B_2 \\ v'(\xi) & \text{otherwise} \end{cases} .$$

Then, for every $\psi \in L(C^1)$,

$$v'(\psi) \in D_1 \oplus D_2 \ \text{ if and only if } \ v(\psi) \in D_1.$$

Therefore $v(\Gamma) \subseteq D_1$ and then $v(\varphi) \in D_1$. From this we get $v'(\varphi) \in D_1 \oplus D_2$. This shows that $\Gamma \models_{M_1^{\mathsf{a}}} \varphi$. ◁

Definition 9.3.7 Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (with $i = 1, 2$) be two matrix logics as in Definition 9.3.5. The *plain fibring* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is the pair

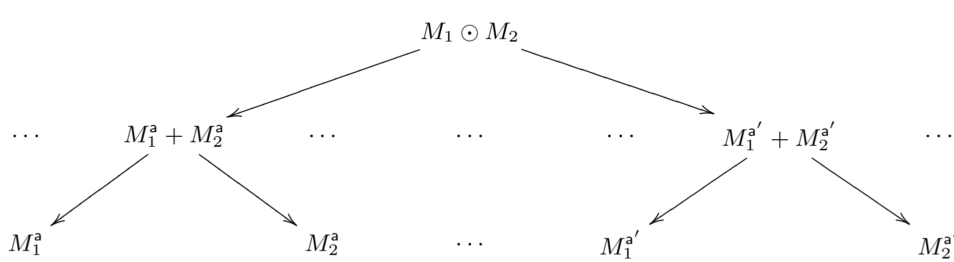
$$\mathcal{C}_1 \odot \mathcal{C}_2 = \langle C^1 \oplus C^2, \models_{M_1 \odot M_2} \rangle$$

such that $M_1 \odot M_2$ is the set of matrices over $C^1 \oplus C^2$,

$$M_1 \odot M_2 = \{M_1^{\mathsf{a}} + M_2^{\mathsf{a}} \ : \ \mathsf{a} \text{ is admissible}\}.$$

▽

The following picture portrays a typical plain fibring configuration.



Now we prove that, in every normal situation, the plain fibring is a conservative extension of its factors. The following is a formal definition of "normal situation".

Definition 9.3.8 The consequence systems $\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be *compatible* if there exist admissible pairs in $B_1^{B_2} \times B_2^{B_1}$. ▽

It is immediate that $\mathcal{C}_1$ and $\mathcal{C}_2$ are compatible if and only if:

(i) $D_1 \neq \emptyset$ if and only if $D_2 \neq \emptyset$, and

(ii) $B_1 \backslash D_1 \neq \emptyset$ if and only if $B_2 \backslash D_2 \neq \emptyset$.

A fundamental property of plain fibring is that the obtained consequence system is a conservative extension of the given consequence systems, provided they are compatible.

Proposition 9.3.9 *Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (with $i = 1, 2$) be two matrix logics as in Definition 9.3.5 such that $\mathcal{C}_1$ and $\mathcal{C}_2$ are compatible. Then $\mathcal{C}_1 \odot \mathcal{C}_2$ is a conservative extension of both $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Proof. We just prove that $\mathcal{C}_1 \odot \mathcal{C}_2$ is a conservative extension of $\mathcal{C}_1$, because the proof for $\mathcal{C}_2$ is analogous.

Thus, let $\Gamma \cup \{\varphi\} \subseteq L(C^1)$. Then $\Gamma \models_{M_1} \varphi$ if and only if $\Gamma \models_{M_1^{\mathsf{a}}} \varphi$ for every admissible pair a, by Proposition 9.3.6 (observe that the compatibility assumption is used in order to prove the 'if' part), if and only if $\Gamma \models_{M_1^{\mathsf{a}}+M_2^{\mathsf{a}}} \varphi$ for every admissible pair a (by Proposition 9.3.3), if and only if $\Gamma \models_{M_1 \odot M_1} \varphi$. ◁

Example 9.3.10 Recall the 3-valued matrix logic I^1 described in Example 9.2.9. Let $\mathcal{C}_1$ be the 3-valued consequence system for the negation $\neg$ of I^1 defined by the matrix M_1 and let $\mathcal{C}_2$ be the 2-valued consequence system for the classical implication $\Rightarrow$ defined over *2* defined by the matrix M_2. The corresponding truth-tables are displayed below.

	$\neg$
T	F
F_1	F
F	T

$\Rightarrow$	1	0
1	1	0
0	1	1

Note that $D_1 = \{T\}$ and $D_2 = \{1\}$ are the sets of designated values in M_1 and M_2, respectively, where $B_1 = \{T, F_1, F\}$ and $B_2 = \{1, 0\}$ are the respective domains.

Let $B = \{T, T_1, F, 1, 0\}$ and $D = \{T, T_1, 1\}$ be the disjoint union of B_1 and B_2 and of D_1 and D_2, respectively. Consider the pair $\mathsf{a} = (f_1, f_2)$ in $B_1^{B_2} \times B_2^{B_1}$ such that

- $f_1(1) = T$ and $f_1(0) = F$;
- $f_2(T) = 1$ and $f_2(F_1) = f_2(F) = 0$.

Clearly the pair a is admissible and the matrices M_1^{a} and M_2^{a} are given by the following truth-tables.

	$\neg$
T	F
1	F
F_1	F
F	T
0	T

$\Rightarrow$	T	1	F_1	F	0
T	1	1	0	0	0
1	1	1	0	0	0
F_1	1	1	1	1	1
F	1	1	1	1	1
0	1	1	1	1	1

Let $\mathcal{C}$ be the consequence system defined over the signature $\{\neg, \Rightarrow\}$ and characterized by the matrix $M_1^{\mathsf{a}} + M_2^{\mathsf{a}}$ given by the two tables above, where $\{T, 1\}$ is the set of designated values. The truth-values T and 1 are congruent, and F and 0 are also congruent. Therefore the reduced matrix for $\mathcal{C}$ coincides with the 3-valued logic I^1 presented in Example 9.2.9. $\triangledown$

The last example shows that the direct union of matrices, together with the extension of matrices, can help to better understand some matrix logics (in [48] the 3-valued paraconsistent logic P^1 is recovered from a 3-valued negation and classical 2-valued implication in a similar way). The next example shows what happens with the plain fibring in the situation described in the example above.

Example 9.3.11 With the same notation as above consider, given M_1 and M_2, another admissible pair $\mathsf{a}' = (g_1, g_2)$ such that $g_2 = f_2$ but $g_1(1) = T$ and $g_1(0) = F_1$. It is easy to see that a and a' are the unique admissible pairs. Note that $M_2^{\mathsf{a}'} = M_2^{\mathsf{a}}$. On the other hand, $M_1^{\mathsf{a}'}$ is defined as follows:

	$\neg$
T	F
1	F
F_1	F
F	T
0	F

From this it follows that the consequence system $\mathcal{C}_1 \odot \mathcal{C}_2$, the plain fibring of $\mathcal{C}_1$ and $\mathcal{C}_2$, is characterized by the set of matrices

$$M_1 \odot M_2 = \{M_1^{\mathsf{a}} + M_2^{\mathsf{a}},\ M_1^{\mathsf{a}'} + M_2^{\mathsf{a}'}\}.$$

In Example 9.3.10 it was observed that $M_1^{\mathsf{a}} + M_2^{\mathsf{a}}$ defines the logic I^1 and then, for instance,

$$\models_{M_1^{\mathsf{a}}+M_2^{\mathsf{a}}} ((\neg(\neg(\xi_1 \Rightarrow \xi_2))) \Rightarrow (\xi_1 \Rightarrow \xi_2)).$$

By its turn, in $M_1^{\mathsf{a}'} + M_2^{\mathsf{a}'}$ it can proved that

$$\not\models_{M_1^{\mathsf{a}'}+M_2^{\mathsf{a}'}} ((\neg(\neg(\xi_1 \Rightarrow \xi_2))) \Rightarrow (\xi_1 \Rightarrow \xi_2)).$$

In order to prove this, it is enough to consider any valuation v over the matrix $M_1^{\mathsf{a}'} + M_2^{\mathsf{a}'}$ such that $v(\xi_1) \in \{T, 1\}$ and $v(\xi_2) \in \{F_1, F, 0\}$. Then $v(\xi_1 \Rightarrow \xi_2) = 0$ and so

$$v(\neg(\neg(\xi_1 \Rightarrow \xi_2))) = (\neg(\neg 0)) = (\neg F) = T.$$

From this it follows that

$$v((\neg(\neg(\xi_1 \Rightarrow \xi_2))) \Rightarrow (\xi_1 \Rightarrow \xi_2)) = (T \Rightarrow 0) = 0.$$

Therefore $((\neg(\neg(\xi_1 \Rightarrow \xi_2))) \Rightarrow (\xi_1 \Rightarrow \xi_2))$ is not a theorem of the consequence system characterized by $M_1^{\mathsf{a}'} + M_2^{\mathsf{a}'}$, and so

$$\not\models_{M_1 \odot M_2} ((\neg(\neg(\xi_1 \Rightarrow \xi_2))) \Rightarrow (\xi_1 \Rightarrow \xi_2)).$$

∇

Example 9.3.12 In [93], the hierarchy $\{I^n\}_{n\in\mathbb{N}}$ of weakly-intuitionistic logics generalizing I^1 was introduced. For every n, the logic I^n is defined over the signature C^{I^n} with symbols $\{\neg_{I^n}, \Rightarrow_{I^n}\}$, and its semantics is given by the matrix

$$M_{I^n} = \langle \mathcal{B}_{I^n}, \{\mathbf{t}\}\rangle$$

such that $B_{I^n} = \{\mathbf{t}, F_0, F_1, \ldots, F_n\}$. The operations of the matrix M_{I^n} are given by the tables below, where $1 \leq l \leq n$.

	$\neg_{I^n}$
$\mathbf{t}$	F_0
F_0	$\mathbf{t}$
F_l	F_{l-1}

$\Rightarrow_{I^n}$	$\mathbf{t}$	F_0	F_l
$\mathbf{t}$	$\mathbf{t}$	F_0	F_0
F_0	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
F_l	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$

The 3-valued paraconsistent logic P^1 was introduced in [244] and it is, in a certain sense, dual to I^1 (recall that the logic P^1 is also an **LFI**, as it was mentioned in Chapter 5). Also in [93] it was introduced a hierarchy of paraconsistent logics, $\{P^n\}_{n\in\mathbb{N}}$, which generalizes the logic P^1. Each P^n is defined over the signature C^{P^n} with symbols $\{\neg_{P^n}, \Rightarrow_{I^n}\}$ and whose semantics is given by the matrix

$$M_{P^n} = \langle \mathcal{B}_{P^n}, \{T_0, T_1, \ldots, T_n\}\rangle$$

such that $B_{P^n} = \{T_0, T_1, \ldots, T_n, \mathbf{f}\}$. The corresponding operations are displayed in the tables below, where $1 \leq h \leq n$.

	$\neg_{P^n}$
T_0	$\mathbf{f}$
T_h	T_{h-1}
$\mathbf{f}$	T_0

$\Rightarrow_{P^n}$	T_0	T_h	$\mathbf{f}$
T_0	T_0	T_0	$\mathbf{f}$
T_h	T_0	T_0	$\mathbf{f}$
$\mathbf{f}$	T_0	T_0	T_0

It is worth noting that both P^0 and I^0 coincide with classical propositional logic over the connectives $\{\neg, \Rightarrow\}$ with 2-valued matrix semantics.

In this example the plain fibring of I^n with P^k will be analyzed. We begin by observing that, given I^n and P^k, every admissible pair has the form $\mathsf{a}_{ij} = (g_i, f_j)$, for $0 \leq j \leq k$ and $0 \leq i \leq n$, such that

- $g_i(\mathbf{f}) = F_i$ and $g_i(T_h) = \mathbf{t}$, for $0 \leq h \leq k$;
- $f_j(\mathbf{t}) = T_j$ and $f_j(F_l) = \mathbf{f}$, for $0 \leq l \leq n$.

Let $M_{ij} = M_{I^n}^{\mathsf{a}_{ij}} + M_{P^k}^{\mathsf{a}_{ij}}$. Then the matrix M_{ij} is defined over the signature C^{nk} with set of connectives

$$\{\neg_{In}, \Rightarrow_{In}, \neg_{Pk}, \Rightarrow_{Pk}\}.$$

The domain of M_{ij} is

$$\{\mathbf{t}, T_0, T_1, \ldots, T_k, F_0, F_1, \ldots, F_n, \mathbf{f}\}$$

and the set of designated values is $\{\mathbf{t}, T_0, T_1, \ldots, T_k\}$. The operations are given below, observing that the truth-table of the negation $\neg_{In}^i$ must consider, separately, the cases $i = 0$ and $i > 0$; analogously, the truth-table of the negation $\neg_{Pk}^j$ must consider the cases $j = 0$ and $j > 0$. In the tables below, $1 \leq h \leq k$ and $1 \leq l \leq n$.

$\Rightarrow_{In}^i$	$\mathbf{t}$	T_0	T_h	F_0	F_l	$\mathbf{f}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	F_0	F_0	F_0
T_0	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	F_0	F_0	F_0
T_h	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	F_0	F_0	F_0
F_0	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
F_l	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$

$\Rightarrow_{Pk}^j$	$\mathbf{t}$	T_0	T_h	F_0	F_l	$\mathbf{f}$
$\mathbf{t}$	T_0	T_0	T_0	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
T_0	T_0	T_0	T_0	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
T_h	T_0	T_0	T_0	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
F_0	T_0	T_0	T_0	T_0	T_0	T_0
F_l	T_0	T_0	T_0	T_0	T_0	T_0
$\mathbf{f}$	T_0	T_0	T_0	T_0	T_0	T_0

	$\mathbf{t}$	T_0	T_h	F_0	F_l	$\mathbf{f}$
$\neg_{In}^0$	F_0	F_0	F_0	$\mathbf{t}$	F_{l-1}	$\mathbf{t}$
$\neg_{In}^i$	F_0	F_0	F_0	$\mathbf{t}$	F_{l-1}	F_{i-1}
$\neg_{Pk}^0$	$\mathbf{f}$	$\mathbf{f}$	T_{h-1}	T_0	T_0	T_0
$\neg_{Pk}^j$	T_{j-1}	$\mathbf{f}$	T_{h-1}	T_0	T_0	T_0

The matrix logic characterized by M_{ij} is simultaneously paraconsistent (with respect to the negation $\neg_{P_k}$) and weakly-intuitionistic (with respect to the negation $\neg_{I_n}$). On the other hand, the plain fibring $I^n \odot P^k$ of I^n and P^k is the matrix logic characterized by the set of matrices

$$M_{I^n} \odot M_{P^k} = \{M_{ij} \ : \ 0 \leq i \leq n \text{ and } 0 \leq j \leq k\}.$$

$\triangledown$

As mentioned above, plain fibring can be seen as a kind of fibring by functions. This relationship is suggested by a characterization of plain fibring through valuations, which resembles fibring by functions as defined in Subsection 1.2.3 of Chapter 1.

Definition 9.3.13 Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (for $i = 1, 2$) be two matrix logics, where each M_i is a matrix over C^i with domain B_i and set of designated values D_i. A *fibred valuation* is a triple

$$(f_1, f_2, v)$$

such that $(f_1, f_2) \in B_1^{B_2} \times B_2^{B_1}$ is an admissible pair and $v : \Xi \to B_1 \oplus B_2$ is a mapping. Given $\varphi \in L(C^1 \oplus C^2)$ and a fibred valuation $w = (f_1, f_2, v)$, we define $w(\varphi) \in B_1 \oplus B_2$ by recursion on the complexity of φ as follows:

- $w(\xi) = v(\xi)$ for $\xi \in \Xi$;
- $w(c(\varphi_1, \ldots, \varphi_k)) = c(\widetilde{w}(\varphi_1), \ldots, \widetilde{w}(\varphi_k))$ for $c \in (C^1 \oplus C^2)_k$ and $\varphi_1, \ldots, \varphi_k \in L(C^1 \oplus C^2)$ where, for every formula φ:

$$\widetilde{w}(\varphi) = \begin{cases} w(\varphi) & \text{if } c \in C^i_k \text{ and } w(\varphi) \in B_i, \text{ for } i = 1, 2 \\[2ex] f_1(w(\varphi)) & \text{if } c \in C^1_k \text{ and } w(\varphi) \in B_2 \\[2ex] f_2(w(\varphi)) & \text{if } c \in C^2_k \text{ and } w(\varphi) \in B_1 \end{cases}$$

$\triangledown$

We say that a fibred valuation w *satisfies* φ if $w(\varphi) \in D_1 \oplus D_2$. The *plain fibred consequence relation* $\models_{M_1 \otimes M_2}$ over $L(C^1 \oplus C^2)$ is defined as follows:

$$\Gamma \models_{M_1 \otimes M_2} \varphi$$

if every fibred valuation w satisfying simultaneously all the formulas of Γ also satisfies φ.

In [60] it was proved the following:

Proposition 9.3.14 *The pair $\langle C^1 \oplus C^2, \models_{M_1 \otimes M_2} \rangle$ is a consequence system which coincides with the plain fibring $\mathcal{C}_1 \odot \mathcal{C}_2$ of $\mathcal{C}_1$ and $\mathcal{C}_2$.*

From the last result, the analogy between plain fibring and fibring by functions should be clear.

It is worth noting that plain fibring is an operation with the following characteristic: starting from matrix logics $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\mathcal{C}_i$ is characterized by a single matrix M_i (for $i = 1, 2$), the resulting consequence system is a matrix logic characterized by a set of matrices (which, in general, is not a singleton). This asymmetry can be easily avoided by generalizing the operation of plain fibring to matrix logics in general.

Definition 9.3.15 Let $\mathcal{C}_i = \langle C^i, \mathcal{K}_i \rangle$ (with $i = 1, 2$) be two matrix logics where each $\mathcal{K}_i$ is a class of matrices. The *plain fibring of* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ is the matrix logic

$$\mathcal{C}_1 \odot \mathcal{C}_2 = \langle C^1 \oplus C^2, \mathcal{K}_1 \odot \mathcal{K}_2 \rangle$$

such that $\mathcal{K}_1 \odot \mathcal{K}_2$ is the class of matrices over $C^1 \oplus C^2$,

$$\{M_1^{\mathsf{a}} + M_2^{\mathsf{a}} \ : \ M_1 \in \mathcal{K}_1, \ M_2 \in \mathcal{K}_2 \ \text{ and } \ \mathsf{a} \in B_1^{B_2} \times B_2^{B_1} \text{ is admissible}\}.$$

$\triangledown$

It should be clear that, in case $\mathcal{K}_i = \{M_i\}$ (for $i = 1, 2$) then $\mathcal{C}_1 \odot \mathcal{C}_2$ coincides with the plain fibring of $\mathcal{C}_1$ and $\mathcal{C}_2$ introduced in Definition 9.3.7.

Proposition 9.3.16 *With notation as above, suppose that the logics* $\langle C^1, M_1 \rangle$ *and* $\langle C^2, M_2 \rangle$ *are compatible, for every* $M_1 \in \mathcal{K}_1$ *and every* $M_2 \in \mathcal{K}_2$. *Then* $\mathcal{C}_1 \odot \mathcal{C}_2$ *is a conservative extension of both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$.

Proof. Let $\Gamma \cup \{\varphi\} \subseteq L(C^1)$. Then $\Gamma \models_{\mathcal{K}_1} \varphi$ if and only if, for every $M_1 \in \mathcal{K}_1$ and every $M_2 \in \mathcal{K}_2$, $\Gamma \models_{M_1 \odot M_2} \varphi$, by adapting the proof of Proposition 9.3.9, if and only if $\Gamma \models_{\mathcal{K}_1 \odot \mathcal{K}_2} \varphi$. The proof for $\mathcal{C}_2$ is analogous. The details are left to the reader. $\triangleleft$

A matrix $M = \langle \mathcal{B}, D \rangle$ is said to be *trivial* if $D = \emptyset$ or $D = B$. The following result is a direct consequence of the proposition above.

Corollary 9.3.17 *Let* $\mathcal{C}_i = \langle C^i, \mathcal{K}_i \rangle$ *(with* $i = 1, 2$*) be two matrix logics. Suppose that* $\mathcal{K}_1$ *and* $\mathcal{K}_2$ *do not contain trivial matrices. Then* $\mathcal{C}_1 \odot \mathcal{C}_2$ *is a conservative extension of both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$.

We end this chapter by presenting a slight generalization of plain fibring. Recall from Definition 9.3.7 that the plain fibring of M_1 and M_2 is characterized by the set of matrices

$$M_1 \odot M_2 = \{M_1^{\mathsf{a}} + M_2^{\mathsf{a}} \ : \ \mathsf{a} \text{ is admissible}\}.$$

It is natural to investigate the case where there are no restrictions to the pairs a. This lead us to the following definition.

Definition 9.3.18 Let $\mathcal{C}_i = \langle C^i, M_i \rangle$ (with $i = 1, 2$) be two matrix logics as in Definition 9.3.5. The *unrestricted plain fibring* of $\mathcal{C}_1$ and $\mathcal{C}_2$ is the pair

$$\mathcal{C}_1 \circledast \mathcal{C}_2 = \langle C^1 \oplus C^2, \models_{M_1 \circledast M_2} \rangle$$

such that $M_1 \circledast M_2$ is the set

$$M_1 \circledast M_2 = \{M_1^{\mathsf{a}} + M_2^{\mathsf{a}} \ : \ \mathsf{a} \in B_1^{B_2} \times B_2^{B_1}\}$$

of matrices over $C^1 \oplus C^2$. ▽

Surprisingly enough, the unrestricted plain fibring of $\mathcal{C}_1$ and $\mathcal{C}_2$ is a weak conservative extension of both $\mathcal{C}_1$ and $\mathcal{C}_2$ (see Proposition 9.3.20 below). In order to prove this, we need the following lemma.

Lemma 9.3.19 *Let* $\mathsf{a} = (f_1, f_2)$ *in* $B_1^{B_2} \times B_2^{B_1}$ *and let* v' *be a* $(M_1^{\mathsf{a}} + M_2^{\mathsf{a}})$*-valuation. Consider the* M_1*-valuation* $v : L(C^1) \to B_1$ *such that*

$$v(\xi) = \begin{cases} f_1(v'(\xi)) & \text{if } v'(\xi) \in B_2 \\ \\ v'(\xi) & \text{otherwise} \end{cases}$$

for every $\xi \in \Xi$*. Then* $v(\varphi) = v'(\varphi)$ *for every* $\varphi \in L(C^1) \setminus \Xi$*. An analogous result holds for* M_2 *(using* f_2 *instead of* f_1*).*

Proof. Straightforward, by induction on the complexity of φ and Definition 9.3.5. ◁

Proposition 9.3.20 *Let* $\mathcal{C}_i$ *be a non-trivial consequence system induced by a matrix* M_i*,* $i = 1, 2$*. Then* $\mathcal{C}_1 \circledast \mathcal{C}_2$ *is a weak conservative extension of both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$*, that is, for* $i = 1, 2$*:*

$$\models_{M_i} \varphi \ \text{ if and only if } \ \models_{M_1 \circledast M_2} \varphi, \text{ for every } \ \varphi \in L(C^i).$$

Proof. Recall, from Definition 4.1.25 of Chapter 4 that a consequence system is said to be trivial if $\Gamma \vdash \varphi$ for every set of formulas $\Gamma \cup \{\varphi\}$.

By hypothesis, no schema variable is a tautology of $\mathcal{C}_i$ ($i = 1, 2$). In fact, if $\models_{M_i} \xi$ for some schema variable ξ then, by structurality, $\models_{M_i} \varphi$ for every formula φ and so, by monotonicity, $\Gamma \models_{M_i} \varphi$ for every $\Gamma \cup \{\varphi\}$, which contradicts the non-triviality of $\mathcal{C}_i$.

Thus, suppose that $\varphi \in L(C^1)$ is such that $\models_{M_1} \varphi$. Let $\mathsf{a} = (f_1, f_2)$ in $B_1^{B_2} \times B_2^{B_1}$ and let v' be a $(M_1^{\mathsf{a}} + M_2^{\mathsf{a}})$-valuation. Since $\varphi \notin \Xi$, there exists a M_1-valuation v such that

$$v'(\varphi) = v(\varphi)$$

by Lemma 9.3.19. But φ is a M_1-tautology and so $v(\varphi) \in D_1$. That is, $v'(\varphi) \in D_1 \oplus D_2$ and then $\models_{M_1 \circledast M_2} \varphi$.

Conversely, let $\varphi \in L(C^1)$ such that $\models_{M_1 \circledast M_2} \varphi$, and let v be a M_1-valuation. Observe that, by hypothesis, both B_1 and B_2 are non-empty, so $B_1^{B_2} \times B_2^{B_1} \neq \emptyset$. Thus, let $\mathsf{a} = (f_1, f_2)$ in $B_1^{B_2} \times B_2^{B_1}$ and let v' be a valuation over $M_1^{\mathsf{a}} + M_2^{\mathsf{a}}$ such that

$$v'(\xi) = v(\xi)$$

for every $\xi \in \Xi$. By the definition of $M_1^{\mathsf{a}} + M_2^{\mathsf{a}}$ it is easy to prove that

$$v'(\psi) = v(\psi) \quad \text{for every } \psi \in L(C^1).$$

Thus $v'(\varphi) = v(\varphi)$ and, since $\models_{M_1 \circledast M_2} \varphi$, then $v(\varphi) \in D_1$. Therefore, $\models_{M_1} \varphi$. The proof for $\mathcal{C}_2$ is analogous. ◁

Note that, if exactly one of the consequence systems (say, $\mathcal{C}_1$) is trivial, then the last result is no longer true. In fact: assuming that $B_1 = D_1 \neq \emptyset$ and $\mathcal{C}_2$ is not trivial then the schema variable ξ_1 is a $\mathcal{C}_1$-tautology but not a $\mathcal{C}_2$-tautology. Since $B_1^{B_2} \times B_2^{B_1} \neq \emptyset$, take $\mathsf{a} \in B_1^{B_2} \times B_2^{B_1}$. It is easy to define a $(M_1^{\mathsf{a}} + M_2^{\mathsf{a}})$-valuation v' such that $v'(\xi_1) \in B_2 \setminus D_2$. Thus,

$$v'(\xi_1) \notin D_1 \oplus D_2$$

and so $\not\models_{M_1 \circledast M_2} \xi_1$, despite $\models_{M_1} \xi_1$.

The Proposition 9.3.20 cannot be improved. In fact, $\mathcal{C}_1 \circledast \mathcal{C}_2$ is not, in general, a strong extension of the given logics, as the following example shows.

Example 9.3.21 Consider the disjunctive fragment of the classical propositional logic

$$\mathcal{C}_1 = \langle C^1, \models_1 \rangle$$

such that $C_2^1 = \{\vee\}$ and $C_k^1 = \emptyset$ in any other case, induced by the matrix $M_1 = \langle \{0,1\}, \{1\} \rangle$ with its usual truth-table. On the other hand, consider any consequence system

$$\mathcal{C}_2 = \langle C^2, \models_2 \rangle$$

such that $\models_2$ is defined by a matrix

$$M_2 = \langle \{T, T_1, F\}, \{T, T_1\} \rangle.$$

The signature C^2 and the operations of M_2 are irrelevant here.

Obviously, $\xi_1 \models_1 \xi_1 \vee \xi_2$. Now, let $f_1 : \{T, T_1, F\} \to \{0,1\}$ such that $f_1(T) = 0$, and let f_2 be any mapping $f_2 : \{0,1\} \to \{T, T_1, F\}$. Consider now $\mathsf{a} = (f_1, f_2)$ and define the $(M_1^{\mathsf{a}} + M_2^{\mathsf{a}})$-valuation v such that

$$v(\xi_1) = T \text{ and } v(\xi_2) = 0.$$

Then $v(\xi_1) \in D_1 \oplus D_2$. On the other hand,

$$v(\xi_1 \vee \xi_2) = (v(\xi_1) \vee v(\xi_2)) = (f_1(T) \vee 0) = 0 \vee 0 = 0 \notin D_1 \oplus D_2.$$

Hence, $\xi_1 \not\models_{M_1 \circledast M_2} \xi_1 \vee \xi_2$. ∇

Of course an analogous generalization of Definition 9.3.15, Proposition 9.3.16 and Corollary 9.3.17 can be obtained for unrestricted plain fibring. In this case, the unrestricted plain fibring of two matrix logics $\mathcal{C}_i = \langle C^i, \mathcal{K}_i \rangle$ (with $i = 1, 2$), such that $\mathcal{K}_1$ and $\mathcal{K}_2$ do not contain trivial matrices, is a weak conservative extension of the given logics. The details of these constructions are left to the reader.

9.4 Final remarks

This chapter addresses the question of combining logics from the point of view of decomposition (splitting), which is dual to the point of view of composition (splicing), studied in the previous chapters.

Up to now, we have exclusively analyzed combination of logics from the bottom-up perspective of splicing logics; in particular, we have restricted ourselves to the study of the technique of fibring. The top-down perspective of splitting logics deserves equal attention, as the potentialities for its applications are really significant: the examples and results obtained in this chapter are an evidence of this claim.

One interesting consequence of the adopted point of view of splitting logics can be appreciated from the categorical perspective: the notion of signature morphism can be enlarged, obtaining a new signature category called **sSig**. In **sSig**, a k-ary connective c can be mapped by a morphism f into a formula $\varphi(\xi_1, \ldots, \xi_k)$ containing exactly the schema variables $\xi_1, \ldots, \xi_k$. Thus, the extended function $\widehat{f}$ associated to f assigns, to any formula with k schema variables, a formula containing exactly the same schema variables.

The new morphisms of **sSig**, when considered in the associated category **sCon** of consequence systems, allow us to define translations which are closer to the translations between logic systems occurring in real examples. For instance, two presentations of propositional classical logic over different signatures can be inter-translated as expected. Moreover, this new category of signatures has products, which are necessary in order to split a logic system into factors.

In this chapter, two particular semantical methods for splitting logics were analyzed: possible-translations semantics in Section 9.2, and direct union of matrices and the related plain fibring in Section 9.3.

Possible-translations semantics have been shown to be a very general method, embodying matrix semantics. Clearly, the generality of PTS's amply justifies the existence of more restrained methods as direct union of matrices and plain fibring.

Finally, the decomposition of a logic into simpler components could offer additional tools for attacking problems of complexity of algorithms (via the satisfiability problem), questions in proof-theory and in algebraization of logics. Additionally, it makes sense to define and to characterize which are the prime logics of a given logic: the ones that cannot be further split (up to a given method). For instance, matrix logics defined over a signature containing just one connective could be considered as prime logics with respect to direct union of matrices. The study of a logic from its prime factors may constitute a new and interesting standpoint to the theory of general consequence systems.

Chapter 10

New trends: Network fibring

In the previous chapters we investigated different aspects of fibring, always in a clear logical context. That is, we always assumed that the original components were logics with certain characteristics. Now we are faced with a new problem. There are application domains where we seem to have a fibring situation but the problem is initially presented with networks instead of logic systems. One of the messages is that if we look at the problem with networks seen as labeled deductive systems then we immediately are in a logical context. In this way, we broaden substantially the applications of fibring to many unusual domains like neural networks in bioinformatics and argumentation theory. The chapter starts with four case studies illustrating several areas to which we want to extend fibring. In many cases recursive networks are used. The notion of self-fibring of networks is a general abstraction where we can accommodate these different fibring situations.

In Section 10.1, we motivate network fibring using a labelled formulation of modal logic. Next, in Sections 10.2, 10.3, 10.4 and 10.5, we introduce some case-studies. Section 10.2 discusses integration of information flows and describes a system in which reasoning and proofs from different sources of information can be accommodated. In Section 10.3, we refer to some generalizations of logic input/output operations. We also discuss how to combine input/output operations into networks. In Section 10.4, we discuss the fibring of neural networks. In Section 10.5, we turn our attention to recursive Bayesian networks. In Section 10.6, we give the notion of self-fibring of networks. Finally, Section 10.7 presents some concluding remarks.

This chapter capitalizes on the following works: [111] for network modalities, [110] for integrating flows of information using LDS, [79] for neural networks and [275] for Bayesian networks.

10.1 Introduction

The applications of fibring seem to go beyond the case where we want to combine two logics. Examples can be found in such contexts as bioinformatics and argumentation theory. It seems that the most general concept that can be used as the framework for fibring are networks seen as labeled deductive systems [105]. The theory of labeled deductive systems (LDS) was developed from the bottom up point of view, especially to model aspects of human behavior, reasoning and action, and is very comprehensive, adaptable and incremental. It contains a large variety of existing logical systems as special cases. LDS is not a single system but a methodology for building families of systems, ready to be adapted to the needs of various application areas.

In this section, we motivate network fibring by looking essentially at a labeled formulation of modal logic with language L. Herein, we assume that the semantics is presented by a Kripke structure of the form

$$m = \langle W, R, w, \Vdash \rangle$$

where, besides the non-empty set of possible worlds W and the accessibility relation $R \subseteq W^2$, we consider an actual world $w \in W$ and a satisfaction relation $\Vdash \subseteq W \times L$. We consider a satisfaction relation instead of a valuation because we want to stress that the exact recursive definition of satisfaction is not relevant to the discussion. We have that

$$m \Vdash \varphi \;\text{ whenever }\; w \Vdash \varphi.$$

Given a class M of models we can define a semantic consequence relation as follows. We say that a formula γ *entails* a formula φ and write

$$\gamma \vDash \varphi$$

if $m \Vdash \varphi$ for every model $m \in M$ such that $m \Vdash \gamma$.

For the purposes that we have in mind, we need to extend the signature with labels. Labels represent the worlds at the syntactic level. We also need a way to represent R, the accessibility relation, at the syntactic level.

As a consequence, besides the usual modal formulas we also have formulas relating the labels like aRt or $\langle a, t\rangle$, where a and t are labels and labeled formulas like $a : \varphi$ meaning that we want to state that φ is true at a.

A *network specification* is a triple

$$\langle S, F, f\rangle$$

where:

- S is a set of labels (representing worlds at the syntactic level);
- F is a set of formulas of the kind $\langle t_1, t_2\rangle$ (representing the accessibility relation at the syntactic level);

- f is a map that assigns to each label a set of modal formulas.

Note that a network specification can be seen as a (oriented) graph G plus a map f. The labels in S correspond to the nodes of G and the formulas in F correspond to the arrows of G. In the sequel, as usual, G_0 and G_1 denote respectively the set of nodes and the set of arrows of G.

Given a network specification $\Delta = \langle G, f \rangle$, we use $x : \varphi$ to state that $\varphi \in f(x)$. We also write $\Delta(x : \varphi)$ to mean that $x : \varphi$ appears in Δ, that is, $x \in G_0$ and $\varphi \in f(x)$.

Example 10.1.1 An example of a network specification is

$$\Delta = \langle G, f \rangle$$

where

- G is a graph with $G_0 = \{t_1, t_2\}$ and $G_1 = \{\langle t_1, t_2 \rangle\}$;
- f is such that $f(t_1) = \{\varphi\}$ and $f(t_2) = \{\psi\}$.

This specification can be represented in a diagram as in Figure 10.1.

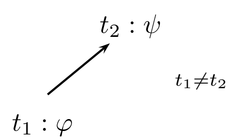


Figure 10.1: Example of a network specification

For simplicity, we can also write the network above as follows:

$$\{t_1, t_2, \langle t_1, t_2 \rangle, t_1 : \varphi, t_2 : \psi\}.$$

∇

We now turn to the semantics of a network specification.

A Kripke structure $\langle W, R, w, \Vdash \rangle$ *satisfies* a network specification $\Delta = \langle G, f \rangle$, denoted by

$$\langle W, R, w, \Vdash \rangle \Vdash \Delta$$

if there is a mapping $g : G_0 \to W$ such that $g(t_1)Rg(t_2)$ whenever $\langle t_1, t_2 \rangle \in G_1$, and $g(t) \Vdash \varphi$ whenever $\varphi \in f(t)$. Sometimes we use $\langle W, R, w, \Vdash \rangle \Vdash_g \Delta$ to indicate that $g : G_0 \to W$ satisfies the above conditions.

We can now define consequence between such network specifications. Network specification Δ_1 *entails* network specification Δ_2, denoted by

$$\Delta_1 \vDash \Delta_2$$

if for every $\langle W, R, w, \Vdash \rangle$ and $g : G_0^1 \to W$ such that $\langle W, R, w, \Vdash \rangle \Vdash_g \Delta_1$ there is $g' : G_0^2 \to W$ such that $g'(x) = g(x)$ for all $x \in G_0^1 \cap G_0^2$ and $\langle W, R, w, \Vdash \rangle \Vdash_{g'} \Delta_2$.

Example 10.1.2 Consider the network specification Δ as described in Figure 10.1. Then, $\{t_1 : (\varphi \wedge (\Diamond((\neg\varphi) \wedge \psi)))\} \vDash \Delta$. ∇

We need to define fibring of network specifications. Having this purpose in mind, we turn now our attention to the cut rule for $\vDash$ as a way to motivate the intended network fibring.

Let Δ and Ω be network specifications. Suppose that we have

- $\Delta(t_1 : \varphi) \vDash \{t_2 : \psi\}$;
- $\Omega \vDash \{t_1 : \varphi\}$.

We would like to substitute $\Omega = \langle G', f' \rangle$ for $t_1 : \varphi$ and get something like $\Delta(\Omega) \vDash t_2 : \psi$. For this we need to define substitution of networks, that is, the substitution of G' for $x \in G$, denoted by $G(x/G')$. This is best explained through an example.

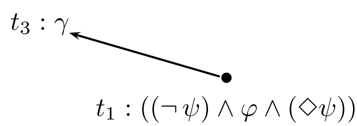


Figure 10.2: Network specification Ω

Example 10.1.3 Let network specification Δ be the one in Example 10.1 and consider the network specification Ω as follows:

$$\{t_1, t_3, \langle t_1, t_3 \rangle, t_1 : ((\neg\,\psi) \wedge \varphi \wedge (\Diamond\psi)), t_3 : \gamma\}$$

(see Figure 10.2). Then we have:

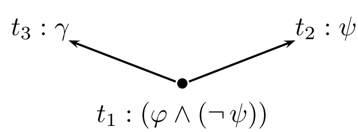


Figure 10.3: Network resulting from substituting Δ into Ω at t_1

- $\Delta \vDash \{t_1 : (\varphi \wedge (\Diamond\psi))\}$
- $\Omega \vDash \{\langle t_1, t \rangle\}$ where $t \neq t_1$.

We would like to perform cut and get possibly

- $\{t_1 : \varphi, t_3 : \gamma, \langle t_1, t_3 \rangle, t_2 : \psi, \langle t_1, t_2 \rangle, t_1 : (\neg\,\psi)\} \vDash \{\langle t_1, t \rangle\}$ where $t \neq t_1$.

Let us look at the network substitutions this requires. We want to substitute Figure 10.1 into Figure 10.2 at point t_1 and get the network specification depicted in Figure 10.3. $\triangledown$

Substitution is easy in a modal logic with an arbitrary accessibility relation. However, how about a modal logic where the relation R is required to satisfy additional conditions? For example in the case of a confluent relation R (that is, if $t_1 R t_2$ and $t_1 R t_3$ then there is t such that $t_2 R t$ and $t_3 R t$, for every t_1, t_2, t_3) the substitution, as the reader can try, is not straightforward. In these cases, we have to do a case analysis and we may have several possible results for the substitution.

10.2 Integrating flows of information

This case study has the following four objectives.

1. To present a modal logical system, slightly more complex than the usual familiar ones, which can serve as an example for new ideas in labeled proof theory, and fibring of networks.
2. To present a system which can model reasoning and proofs from different sources of information, executed at different proof-theoretic levels.
3. To present an example of a system which can *seamlessly* combine a variety of existing logics *at the object level*.
4. To present new ideas and examples about translations of one logic into another.

All of the above points will be realized and exemplified below in a logic which we call $\mathbf{K}_{\text{past}}$. The logic $\mathbf{K}_{\text{past}}$ is a version of past temporal logic with branching tree structure flow of time.

$\mathbf{K}_{\text{past}}$ modal logic

This section concentrates on the modal logic $\mathbf{K}_{\text{past}}$. Before presenting the logic $\mathbf{K}_{\text{past}}$ itself we first introduce some preliminary notions.

In the sequel, we consider propositional Kripke models

$$\langle T, R, w, V \rangle$$

where V is the assignment to the atoms and $w \in T$ is the actual world. We say that x is a predecessor of y whenever xRy. We can assume that the reflexive and transitive closure of R, denoted by $\leq$, is a tree like partial order.

Let $\to$ be strict implication. It is well known that $\to$ satisfies the following satisfaction condition at a world x:

$$x \vDash \varphi \to \psi$$

if $y \vDash \psi$ for all y such that xRy and $y \vDash \varphi$.

Let $(\varphi_1, \ldots, \varphi_n)$ be a sequence of formulas and let ψ be a formula. We define another related kind of satisfaction in a world t. We say that the sequence $(\varphi_1, \ldots, \varphi_n)$ *satisfies* ψ *in* t and write

$$(\varphi_1, \ldots, \varphi_n) \vDash_t \psi$$

whenever the following condition holds: if there exist $t_1, \ldots, t_n$ such that $t_n \vDash \varphi_n$ and, for $1 \leq i \leq n-1$, $t_i \vDash \varphi_i$ and $t_i R\, t_{i+1}$, then $t_n \vDash \psi$.

Observe that $(\varphi_1, \ldots, \varphi_n) \vDash_t \psi$ is satisfaction looking backwards. However, we do have

$$\mathbf{K} \vdash \varphi_1 \to (\varphi_1 \to \ldots \to (\varphi_n \to \psi) \ldots)$$

if $(\varphi_1, \ldots, \varphi_n) \vDash_t \psi$ in every model and every t. If ψ is $(\psi_1 \to \ldots \to (\psi_k \to \delta) \ldots)$ then, using the above definition of satisfaction, we say that

$$(\varphi_1, \ldots, \varphi_n) \vDash_t (\psi_1 \to \ldots \to (\psi_k \to \delta) \ldots)$$

whenever the following condition holds: if $x_1, \ldots, x_n$ and $y_1, \ldots, y_k$ are such that $x_i \vDash \varphi_i$ for $1 \leq i \leq n$, $y_i \vDash \psi_i$ for $1 \leq i \leq k$, and $x_n = t$, then $y_k \vDash \delta$. From this, we get, in particular, the metatheorem of deduction for the logic.

Let $\mathbb{A}_i$ denote a set of formulas. We can write data-structures of the form $(\mathbb{A}_1, \ldots, \mathbb{A}_n)$ and similarly define $(\mathbb{A}_1, \ldots, \mathbb{A}_n) \vDash_t \delta$. The set of formulas of implicational $\mathbf{K}$ is inductively defined as follows:

(i) an atom δ is a formula of level 0;

(ii) if $\mathbb{A}_1, \ldots, \mathbb{A}_n$ are sets of formulas of level less or equal than n and δ is atomic, then $(\mathbb{A}_1, \ldots, \mathbb{A}_n) \to \delta$ is a formula of level less or equal than $n+1$.

Next, we introduce more complex data structures into our logic. We can regard $(\mathbb{A}_1, \ldots, \mathbb{A}_n)$ as a linear tree data-structure τ of the form depicted in Figure 10.4, and write $(\mathbb{A}_1, \ldots, \mathbb{A}_n) \to \varphi$ as $\tau \to \varphi$.

$$x_1 : \mathbb{A}_1 \cdot\!\!\longrightarrow \quad \cdots \quad \longrightarrow\!\!\cdot x_n : \mathbb{A}_n$$

Figure 10.4: Linear tree data-structure

We rewrite the definition of $(\varphi_1, \ldots, \varphi_n) \vDash_t \varphi$ above as follows. We say that τ *satisfies* φ in t and write

$$\tau \vDash_t \varphi$$

whenever the following condition holds: for every order preserving map $g : \tau \mapsto T$ such that $g(x_n) = t$, if $g(x_i) \vDash \mathbb{A}_i$ for all $1 \leq i \leq n$, then $g(x_n) \vDash \varphi$.

We can write $\tau' \to \varphi$. However, what would $\tau' \vDash_t \varphi$ mean? We can use the same definition as above, except that now the order preserving g will embed the nodes of τ' into the subtree below t. This now gives meaning to $\tau' \vDash_t \varphi$.

Can we now continue and give meaning to $t \vDash \tau' \to \delta$? When the tree was a sequence $(x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n) \to \delta$ we understood it as:

$$t \vDash (\mathbb{A}_1, \ldots, \mathbb{A}_n) \to \delta$$

whenever $g(x_n) \vDash \delta$ for all order preserving embeddings g of $(x_1, \ldots, x_n)$ into the *future* of t such that $g(x_i) \vDash \mathbb{A}_i$ for $1 \leq i \leq n$. The embedding of $(x_1, \ldots, x_n)$ into the future of t means that $tR\,g(x_1)$ and $g(x_i)R\,g(x_{i+1})$, for $1 \leq i \leq n-1$.

What would the embedding of τ' into the future of t mean? Do we want to have $tR\,g(x_1)$? Do we want to have $tR\,g(x_2)$? Clearly, we need to indicate in τ' where the input point is supposed to be. We also need to indicate a *top* point where δ is supposed to hold. The way we do this is as follows. We write $(x_1, \ldots, x_n)$ as

$$(e, x_1, \ldots, x_n = a)$$

with e as the input point, and a as the top point. So, embedding this sequence in the future of t means making $g(e) = t$. Thus,

$$(x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n) \to \delta$$

has to be written as

$$(e : \top, x_1 : \mathbb{A}_1, \ldots, x_{n-1} : \mathbb{A}_{n-1}, x_n = a : \mathbb{A}_n) \to \delta.$$

Similarly, $\tau' \to \delta$ is not properly written. We need to say where the input point is. We have several options. Let us choose, for example, the option described in Figure 10.5. Now all is clear. We define the satisfaction of $\tau \to \delta$ at t as follows.

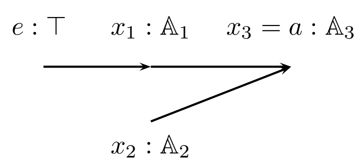


Figure 10.5: Choosing the input point

We say that

$$t \vDash \tau \to \delta$$

whenever the following holds: $g(a) \vDash \delta$ for all order preserving embeddings g of the tree such that $g(x) \vDash \mathbb{A}_x$ for all $x \in \tau'$.

Having gone this far, we can also define recursively trees within trees etc, and define $\vDash$ for them.

The logic described above is ideal for combining logical systems. It was shown in [105] and [114] that many implicational logics such as substructural, strict and intermediate implications, all use modus ponens but differ on the structure of the database and on the procedures governing the use of the rule.

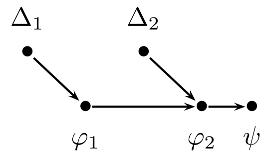


Figure 10.6: Combination of logics

Our data-structures allow us to represent seamlessly combinations of such logics. To illustrate this point, suppose $\Delta_1 \vdash_1 \varphi_1$ in logic 1 and $\Delta_2 \vdash_2 \varphi_2$ in logic 2 and $\varphi_1, \varphi_2 \vdash_3 \psi$ in logic 3. We can represent this situation as in Figure 10.6.

Below we show how concatenation logic can be interpreted in $\mathbf{K}_{\text{past}}$. Since strict implication is part of $\mathbf{K}_{\text{past}}$ already, it shows how the two implications can live together as one! Other logics such as linear implication and even the non-monotonic conditional can also be embedded in our logic $\mathbf{K}_{\text{past}}$.

We now concentrate on the logic $\mathbf{K}_{\text{past}}$. In the sequel, all trees and all sets of trees are always assumed to be finite.

Definition 10.2.1 The set of *trees* is inductively defined as follows:

- a one-point tree is a tree;
- if $\langle T_1, R_1, a_1\rangle, \ldots, \langle T_n, R_n, a_n\rangle$ are trees with top points $a_i \in T_i$, predecessor relations $R_i \subseteq T_i \times T_i$ and such that the sets T_i are pairwise disjoint, then

  $$T = (\{a\} \cup \bigcup T_i, R, a)$$

  where a is a new point and $R = \{(a_i, a)\} \cup \bigcup R_i$, is a tree.

A tree with several *input* points has the form $\langle T, R, a, e_i\rangle$ where $e_i \in T$ are the input points for $1 \leq i \leq n$. We identify e_1 as the main input point. $\triangledown$

Clearly, we say that x *is the predecessor of* y whenever $xR_i\, y$.

Note that in the definition of tree with several input points we may have $n = 1$ or $n = 2$, etc.

Figure 10.7 illustrates a tree with top point a, obtained from trees T_1, T_2,..., T_n with top points a_1, a_2,..., a_n, respectively.

Definition 10.2.2 The *joining of trees with input points* is defined as follows.

Let $\tau = \langle T, R, a, e\rangle$ be a tree with input point e. Let $\tau_i = (T_i, R_i, e)$, for some i, be other trees with the same top point e. Assume that they are all pairwise disjoint except for the common point e. Then the joining is the tree:

$$\{\tau_i\} + \tau = \langle \bigcup T_i, \bigcup R_i, a\rangle.$$

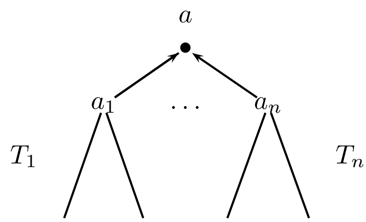


Figure 10.7: Example of a tree

We can have several input points and input trees into them. Consider the tree $\tau = \langle T, R, a, e_i \rangle$, $e_i \in T$, and for each i, let $\tau_{i,j} = \langle T_{i,j}, R_{i,j}, e_i \rangle$ be other pairwise disjoint trees such that $T_{i,j} \cap T = e_i$. Then the joining is the tree:

$$\{\tau_{i,j}\} + \tau = \langle \bigcup T_{i,j}, \bigcup R_{i,j}, a \rangle.$$

Considering the family $\{\tau^k = \langle T^k, R^k, a^k, e_i^k \rangle\}$ of such trees with corresponding $\tau_{i,j}^k$ trees for input, as above, we let

$$\{\tau_{i,j}^k\} + \{\tau^k\} = \{(\{\tau_{i,j}^k\} + \tau^k)\}.$$

∇

Next, we turn our attention to the splitting of trees.

Definition 10.2.3 Let $\tau = \langle T, R, a \rangle$ and $\tau_i = \langle T_i, R_i, t_i \rangle$ be trees and let $\leq_T$ be the transitive and reflexive closure of R (hence, $x \leq_T a$ for every $x \in T$).

Let $t_1, t_2, \ldots, t_n$ be pairwise incomparable points in T with respect to $\leq_T$, that is, if $t_i \leq_T t_j$ then $t_i = t_j$ and consider

- $T_{t_i} = \{x \in T : x \leq t_i\}$ and $R_{t_i} = R \restriction T_{t_i}$;
- $T' = (T \setminus \bigcup_i T_{t_i}) \cup \{t_1, \ldots, t_n\}$ and $R' = R \restriction T'$.

We define

$$\langle T, R, a \rangle = \{\langle T_{t_i}, R_i, t_i \rangle\} + \langle T', R', a \rangle$$

and say that the tree τ *was split at the points* $\{t_1, t_2, \ldots, t_n\}$.

We say τ_i *are ready for input* into τ whenever

- $T_i \cap T = \{t_i\}$;
- t_i are pairwise incomparable with respect to $\leq_T$.

The result of the input is the tree

$$\langle\bigcup_i T_i \cup T, \bigcup_i R_i \cup R, a\rangle = \{\langle T_i, R_i, t_i\rangle + \langle T, R, a\rangle\}.$$

Let $\mathbb{T}_1$ and $\mathbb{T}_2$ be sets of pairwise disjoint trees and assume that for any tree τ from $\mathbb{T}_2$ and τ' from $\mathbb{T}_1$ either they are disjoint or τ' is ready for input into τ. Let $\mathbb{T}_1^\tau = \{\tau' \in \mathbb{T}_1 : \tau' \text{ is ready for input into } \tau\}$ and assume that $\mathbb{T}_1 = \bigcup_{\tau\in\mathbb{T}_2} \mathbb{T}_1^\tau$. Then, we say that $\mathbb{T}_1$ *is ready to input into* $\mathbb{T}_2$ and we define

$$\mathbb{T}_1 + \mathbb{T}_2 = \{\mathbb{T}_1^\tau + \tau\}.$$

∇

If a tree τ is split at several points then it can be joined back by input.

Remark 10.2.4

1. So far in order to perform the joining $\langle T_1, R_1, a\rangle + \langle T_2, R_2, b\rangle$ we need that $T_1 \cap T_2 = \{a\}$. We can allow the input of any tree into any other tree provided there is an input point.

   Thus, the joining $\langle T_1, R_1, a\rangle + \langle T_2, R_2, b, e\rangle$ can be performed, where $e \in T_2$ is the input point. This can be done by renaming the points of T_1 so that we get a disjoint tree and renaming a to e.

2. Let $\mathbb{T}_1, \ldots, \mathbb{T}_n$ be sets of pairwise disjoint trees such that $\mathbb{T}_i$ is ready for input into $\mathbb{T}_{i+1}$, for each $1 \le i \le n-1$. Then $\mathbb{T}_1 + \mathbb{T}_n$ is well defined and is associative. Define $\mathbb{S}_i$ by induction as follows:

   - $\mathbb{S}_n = \mathbb{T}_n$
   - Since $\mathbb{T}_{n-1}$ is ready for input into $\mathbb{T}_n$ we have $\mathbb{T}_{n-1} = \bigcup_{\tau\in\mathbb{T}_n} \mathbb{T}_{n-1}^\tau$ and $\mathbb{T}_{n-1} + \mathbb{T}_n = \{\mathbb{T}_{n-1}^\tau + \tau : \tau \in \mathbb{T}_n\}$. Let $\mathbb{S}_{n-1} = \mathbb{T}_{n-1} + \mathbb{T}_n$. Since $\mathbb{T}_{n-2}$ is ready for input into $\mathbb{T}_{n-1}$, it is also ready for input into $\mathbb{S}_{n-1}$. Similarly $(\mathbb{T}_{n-2} + \mathbb{T}_{n-1})$ is ready for input into $\mathbb{T}_n$. We have

     $$(\mathbb{T}_{n-2} + \mathbb{T}_{n-1}) + \mathbb{T}_n = \mathbb{T}_{n-2} + (\mathbb{T}_{n-1} + \mathbb{T}_n).$$

     Let $\mathbb{S}_i = \mathbb{T}_i + \mathbb{T}_n$. Clearly, $\mathbb{S} = \sum \mathbb{T}_i$ is one big set of trees. If $\mathbb{T}_n$ contains only one tree then $\mathbb{S}_1$ is one tree. ∇

We now define embedding of trees.

Definition 10.2.5 Let $\tau_i = \langle T_i, R_i, a_i, e_i\rangle$ be trees, $i = 1, 2$. A function

$$g : T_1 \to T_2$$

is said to be a *head embedding* (resp. *tail embedding*) of τ_1 into τ_2 if the following conditions hold:

- $g(a_1) = e_2$ (resp. $g(e_1) = e_2$);
- if $xR_1\, y$ then $g(x)R_2\, g(y)$. ∇

We now define the notion of database of level k hand in hand with the concept of formula of level k.

Definition 10.2.6 *Formulas* of level n and *databases* of level $n+1$ are defined as follows.

- A formula of level 0 is an ordinary formula of the language with $\to$, that is,
  - an atom δ is a level 0 formula;
  - if $\mathbb{A}_1, \ldots, \mathbb{A}_n$ are sets of level 0 formulas, so is $(\mathbb{A}_1 + \mathbb{A}_n) \to \delta$, where δ is atomic.
- A database of level 1 is a pair
$$\langle \tau, f \rangle$$
where $\tau = \langle T, R, a, e_0 \rangle$ is a tree and f is a map such that for every $t \in T$, $f(t)$ is a set of formulas of level 0. We indicate the main input point e_0 but allow for more.
- A formula of level $\leq n+1$ is of the form $(\mathbb{A}_1 + \mathbb{A}_n) \to \delta$ where:
  - $\mathbb{A}_i$ is a set of database trees of level $\leq n+1$;
  - $\mathbb{A}_i$ is ready to be input into $\mathbb{A}_{i+1}$;

  (that is, the trees of the databases of $\mathbb{A}_i$ are ready to be input into the trees of the databases of $\mathbb{A}_{i+1}$). We assume $\mathbb{A}_n$ contains a single database. Whenever $\langle \tau_1, f_1 \rangle$ is input into $\langle \tau_2, f_2 \rangle$, we get a database
$$\langle \tau_1 + \tau_2, f_1 \cup f_2 \rangle$$
  where
$$(f_1 \cup f_2)(x) = \begin{cases} f_1(x), & \text{for } x \in \tau_1 \backslash \tau_2 \\ f_2(x), & \text{for } x \in \tau_2 \backslash \tau_1 \\ f_1(x) \cup f_2(x), & \text{for } x \in \tau_1 \cap \tau_2 \end{cases}$$
  We identify $(\mathbb{A}_1 + \mathbb{A}_n) \to \delta$ with any $(\mathbb{A}_1 + \mathbb{A}_j) \to ((\mathbb{A}_{j+1} + \mathbb{A}_n) \to \delta)$.

  We also say that $\mathbb{A}$ is ready for input into $(\mathbb{B}_1 + \mathbb{B}_n) \to \delta$ in case $\mathbb{A}$ is ready for input into $\mathbb{B}_1 + \mathbb{B}_n$. Recall that, according to Remark 10.2.4, $\mathbb{B} = \mathbb{A}_1 + \mathbb{A}_n$ is one big set of databases.
- A database of level $\leq n+1$ is a pair $\langle \tau, f \rangle$, where τ is a tree with input points and, for $t \in \tau$, $f(t)$ is a set of formulas of level $\leq n$. ∇

Given a database $\langle \tau, f \rangle$ we often use $t : \mathbb{A}$ to state that $f(t) = \mathbb{A}$.

Lemma 10.2.7 *Let $B = (\mathbb{A}_1 + \mathbb{A}_n) \to \delta$ be a formula. Then there exists a single database tree τ such that $(\tau \to \delta) = B$.*

Proof. By induction on n. We assumed in Definition 10.2.6 that $\mathbb{A}_n$ must contain a single tree. Since all trees in $\mathbb{A}_{n-1}$ are ready for input into $\mathbb{A}_n$, we get that $\mathbb{S}_{n-1} = \mathbb{A}_{n-1} + \mathbb{A}_n$ is also a single tree, and so on by induction. $\lhd$

Definition 10.2.8 Let $\langle T, R, t, V\rangle$ be a Kripke model, with $t \in T$. We define the notion of *satisfaction of a formula* at t as follows:

- $t \vDash \delta$ if $t \in V(\delta)$, for δ atomic;
- $t \vDash ((\mathbb{A}_1 + \mathbb{A}_n) \to \delta)$ if the following condition holds:

  if τ' and g are such that
  - $\tau' = \langle T', R', a', e', f\rangle$ is the database $\mathbb{A}_1 + \mathbb{A}_n$;
  - g is any tail embedding of $\langle T', R', e'\rangle$ into $\langle T, R, t\rangle$, considering $t \in T$ as the main input point;
  - $g(y) \vDash f(y)$ for every $y \in T$;

  then $g(a') \vDash f(a')$.

Let $\tau' = \langle T', R', a', e', f\rangle$ be a database and φ a formula. We say that

$$\tau' \vDash_t \varphi$$

whenever the following condition holds: if there exists a head embedding g of τ' into $\langle T, R, t\rangle$ such that $g(y) \vDash f(y)$ for all $y \in T$ then $t \vDash \varphi$. $\triangledown$

Example 10.2.9 The trees associated with a formula $(\mathbb{A}_1 + \mathbb{A}_n) \to \delta$ of ordinary implicational **K** constitute the list

$$((e, t_1, \ldots, t_n), f) \to \delta$$

where $f(e) = \top$, $f(t_i) = \mathbb{A}_i$ and e is the input point.

We have $\mathbb{A} \to (\mathbb{B} \to \delta) = (\mathbb{A} + \mathbb{B}) \to \delta$. This shows why we need e. $\triangledown$

Concatenation logic

Herein, we consider concatenation logic with $\to$ only. We start by defining goal directed computation.

Definition 10.2.10 The relation $\vdash_n$ (for $n \geq 0$) is defined as follows:

- $\Delta \vdash_0 \delta$ if $\Delta = \delta$;
- $(\varphi_1, \ldots, \varphi_n) \to \delta, \psi_1, \ldots, \psi_k \vdash_{m+1} \gamma$ if

– δ is γ;
– $(\psi_1, \ldots, \psi_k) = \Delta_1 + \ldots + \Delta_n$;
– $\Delta_i \neq \emptyset$ and $\Delta_i \vdash_m \varphi_i, 1 \leq i \leq n$. ∇

Lemma 10.2.11 *Let* $\varphi = (\varphi_1, \ldots, \varphi_n) \to \delta$. *If*

(i) $\varphi + \Delta_1 + \Theta + \Delta_2 \vdash_m \delta$

(ii) $\varphi + \Delta_1 + \Delta_2 \vdash_{m'} \delta$

then Θ *is empty.*

Proof. Induction on $max(m, m')$.
Base: For $m = 1$ we must have $\varphi = \delta, \Delta_1 = \Theta = \Delta_2 = \emptyset$.
Step: Assuming the lemma holds for m, m' we show the lemma for $m+1, m'+1$. From (i) we get for some non-empty sets $\Gamma_1, \ldots, \Gamma_n$,

- $\Gamma_i \vdash_m A_i$;
- $\Gamma_1 + \ldots + \Gamma_n = \Delta_1 + \Theta + \Delta_2$.

Similarly, $\Gamma'_1 + \ldots + \Gamma'_n = \Delta_1 + \Delta_2$ and $\Gamma'_i \vdash_{m'} A_i$.
If $\Theta \neq \emptyset$ then there exists a first $i = k$ such that:

- $\Gamma_i = \Gamma'_i$ for $i < k$;
- $\Gamma^*_k = \Gamma^*_k + \Theta^*$;
- $\Gamma^{`*}_k \vdash_{m_*} A_i$ and $\Gamma^*_k + \Theta^* \vdash_{m'_*} A_i$;
- $\{m_*, m'_*\}$ are $\{m', m\}$.

Then Γ^*_k is either Γ_k or Γ'_k whichever is shorter. But this is impossible by the induction hypothesis. $\triangleleft$

Lemma 10.2.12 *We have that* $\varphi \vdash \psi$ *and* $\psi \vdash \varphi$ *imply* $\varphi = \psi$.

The proof of the previous lemma is similar to the proof of Lemma 10.2.11.

Corollary 10.2.13 *If* $(\varphi_1, \ldots, \varphi_n \to \delta), \psi_1, \ldots, \psi_k \vdash (\delta_1, \ldots, \delta_r) \to \delta$ *then there is a unique division*

$$\Delta_1, \ldots, \Delta_m, \Delta_{m+1} \ldots, \Delta_n$$

such that $\Delta_1 + \ldots + \Delta_m + \Delta_{m+1} + \ldots + \Delta_n = \psi_1, \ldots \psi_k, \delta_1, \ldots, \delta_r$, *and* $\Delta_i \vdash \varphi_i$. *We can assume* $\Delta_{m+1} = \Theta_1 + \Theta_2$ *with*

- $\Delta_1 + \ldots + \Delta_m + \Theta_1 = \varphi_1, \ldots, \varphi_n$;
- $\Theta_1 + \Delta_{m+1} + \Delta_n = \delta_1, , \ldots, \delta_r$.

Thus, the division $\Delta_1, \ldots, \Delta_m, \Theta_1$ *for* $\Delta \vdash (\delta_1, \ldots, \delta_r \to \delta)$ *is unique.*

We now provide an illustration.

Example 10.2.14 In Figure 10.8 we check that

$$(y \to c) \to \alpha, a \to (b \to c), x \to a \vdash (x, y \to b) \to \alpha.$$

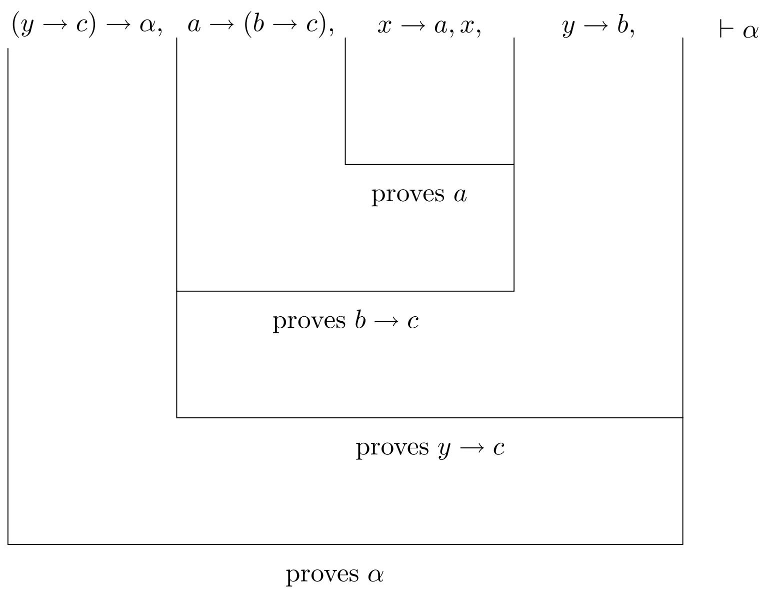


Figure 10.8: Checking $(y \to c) \to \alpha, a \to (b \to c), x \to a \vdash (x, y \to b) \to \alpha$

∇

Example 10.2.15 In order to show that

$$(\beta, x, y \to b) \to \alpha, \beta \vdash (x, y \to b) \to \alpha$$

we check whether

$$(\beta, x, y \to b) \to \alpha, \beta, x, y \to b \vdash \alpha.$$

∇

Translation of concatenation logic into $\mathbf{K}_{\text{past}}$

We can now translate from concatenation logic into $\mathbf{K}_{\text{past}}$. To explain the idea of the translation, we first present some examples.

Consider Example 10.2.14. This is a concatenation logic illustration and we have that when

$$x \to a, x \vdash a$$

the two points $(x \to a)$ and x are replaced by one point. Thus, in concatenation logic we have

$$a \to y, x \to a, a \vdash y.$$

This cannot be done in modal **K** because the points $x \to a$ and x are two possible worlds. We need to shift $a \to y$, that is

$$\top \to (a \to y), x \to a, x \vdash y$$

because $x \to a$ and x can be effectively reduced to $\top$ and a, still two worlds!

However, using our trees, we can simulate the concatenation logic behavior by the tree in Figure 10.9

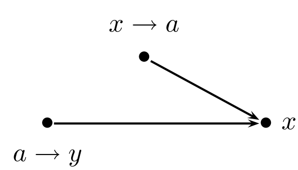


Figure 10.9: Simulating the concatenation logic behavior

We can start with the simple database of Figure 10.10 and input it into the tree of Figure 10.11.

$$(t : x \to a, e : x)$$

Figure 10.10: Simple database

The following is a translation of Example 10.2.14. Let $\tau_2 = \{e : y\}$ and τ_1 be the tree in Figure 10.12. We have that $e \vDash \tau_2 \to c$ because $\tau_1 + \tau_2 \vDash c$. The tree $\tau_1 + \tau_2$ is the tree of Figure 10.12 modified by adding $e : y$ at node e. We have $e \vDash b$ from t_5 and $t_4 \vDash a$ from t_2. Therefore, we have $t_4 \vDash b \to c$ from t_1 and hence $e \vDash c$, as required. Now, since we have $e \vDash \tau_2 \to c$, we can use t_3 and get $e \vDash \alpha$.

$$(s : a \to y, e : \top)$$

Figure 10.11: Tree

We now explain how we got the translation. Our method must be systematic. There are two important remarks. First, note that our translation will be defined only for formulas $(\varphi_1, \ldots, \varphi_n) \to \psi$ such that $(\varphi_1, \ldots, \varphi_n) \vdash \psi$ (that is, we translate theorems only). Second, note that names of nodes in our translation are important because of the definition of $\tau_1 + \tau_2$. The trees of τ_1, τ_2 must share

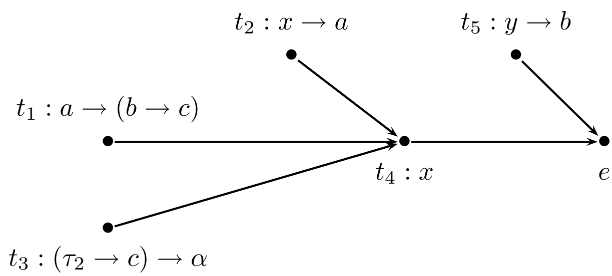


Figure 10.12: Tree τ_1

exactly one node. The translation will make sure that such nodes are well chosen.

Definition 10.2.16 Let $\Delta = (\varphi_1, \ldots, \varphi_n) \vdash_m \psi$ where ψ is $(\psi_1, \ldots, \psi_s) \to \delta$. We translate Δ of concatenation logic into a $\mathbf{K}_{\text{past}}$ database Δ^* and formula ψ^* such that $\Delta^* \vDash \psi^*$. The translation is by induction on m.

- *Case* $m = 0$
  We distinguish two subcases.

  1. ψ_s exists and $\delta = \psi_s$.
     In this case, $\Delta^* = (e : \top)$ and $\psi^* = (e : \delta) \to \delta$. Note that $\varphi_1, \ldots, \varphi_n$ do not exist and $s = 1$.
  2. $\psi_1, \ldots, \psi_s$ do not exist and $n = 1$ and $\varphi_n = \delta$.
     In this case $\Delta^* = \{e : \delta\}$ and $\psi^* = \delta$.

- *Case* $m = 1$
  In this case we have

$$(\delta_1, \ldots, \delta_r) \to \delta, \Delta_1, \ldots, \Delta_{r-k}, \Delta_{r-k+1} = (\Theta_1 + \Theta_2), \Delta_{r-k+1}, \ldots, \Delta_r \vdash_1 \delta$$

  where

  - $((\delta_1, \ldots, \delta_r) \to q, \Delta_1, \ldots, \Delta_{r-k}, \Theta_1) = (\varphi_1, \ldots, \varphi_n)$;
  - $(\Theta_2, \Delta_{r-k+2}, \ldots, \Delta_r) = (\psi_1, \ldots, \psi_s)$.

  We have

$$\begin{array}{l} \Delta_1 \vdash_0 \delta_1 \ldots \\ \vdots \\ \Delta_{r-k} \vdash_0 \delta_{r-k} \\ \Delta_{r-k+1} = \Theta_1 + \Theta_2 \vdash_0 \delta_{r-k+1} \\ \vdots \\ \Delta_r \vdash_0 \delta_r \end{array}$$

The proof theory of concatenation logic does not allow for the Δ_j, and $\Delta_{r-k+1} = \Theta_1 + \Theta_2$ to be empty, so we must have that δ_j are all atomic and are equal to the Δ_j. We can also assume $\Delta_{r-k+1} = \Theta_2 = C_{r-k+1}$.

The following is the $\mathbf{K}_{\text{past}}$ database Δ for proving δ

$$\Delta = ((\delta_1, \ldots, \delta_r) \to \delta, \delta_1, \ldots, \delta_{r-k}, \delta_{r-k+1}, \ldots, \delta_r) \vdash \delta.$$

and it is displayed in Figure 10.13.

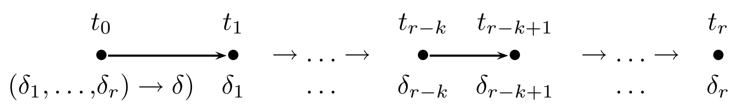


Figure 10.13: Database Δ

The database Δ^* is such that $\Delta^* \vDash \delta$. The way we want to look at this database is as follows. We construct the level 1 databases $\tau_j = (t_j : \delta_j)$, for $j = 1, \ldots, r$. We also construct the database τ of Figure 10.14, where δ_j^* is the translation of δ_j^* and is equal to δ_j.

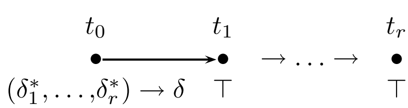


Figure 10.14: Database τ

We now construct the input database

$$\Delta^* = \{\tau_j : j = 1, \ldots, r\} + \tau$$

and get back the original Δ^*.

The above is not enough for our purpose. We need to show a translation of

$$(\varphi_1, \ldots, \varphi_n)^* \vDash ((\psi_1, \ldots, \psi_s) \to \delta)^*$$

However, we can extract that from $\Delta^* \vDash \delta$.

Let

- $\tau' = (t_0 : (\delta_1^*, \ldots, \delta_r^* \to \delta), t_1 : \top, \ldots, t_{r-k} : \delta_{r-k}, t_{r-k+1} : \top)$;
- $(\varphi_1, \ldots, \varphi_n)^* = \tau'$;
- $\tau'' = (t_{r-k+1} : \top, \ldots, t_r : \top)$;
- $(\varphi_1, \ldots, \varphi_n)^* = \{\Delta_j^* : j = 1, \ldots, r - k\} + \tau'$;
- $((\psi_1, \ldots, \psi_s) \to \delta)^* = \{\Delta_j^* : j = r - k + 1, \ldots, r\} + \tau'' \to \delta$.

We have $(\varphi_1, \ldots, \varphi_n)^* \vDash ((\psi_1, \ldots, \psi_s) \to \delta)^*$ because $\Delta^* \vDash \delta$.

- *Case m*

  In order to proceed to case m, we need to formulate the inductive hypothesis carefully. In order to be able to formulate an inductive hypothesis for the translation, we must rigorously describe the situation assumed and the properties of the translation.

  We assume we have

  – $\Delta = (\delta_1, \ldots, \delta_r) \to \delta, \Delta_1, \ldots, \Delta_r \vdash \delta$;

  – $\delta_j = (\delta_1^j, \ldots, \delta_{t_j}^j) \to q_j$, with $\Delta_j \vdash \delta_j, j = 1 \ldots r$.

  A translation is said to be in *standard form* of level m if the following conditions hold:

  There exists a level m database with tree τ as in Figure 10.14 and a set of level 1 trees $\{\tau^\alpha\} = \Delta_j^*$ and level m formulas $\delta_j^* = (\delta_1^j, \ldots, \delta_{t_j}^j)^* \to q_j$ such that Δ_j^* is ready for input into c_j^* and $\Delta_j^* \vDash \delta_j^*$, we have $\Delta^* = \{\Delta_j^*\} + \tau$, and all trees involved are disjoint.

  We also assume that every formula α in Δ_j, such that $\Delta_j = \Theta_j^1 + \Theta_j^2$, with α first formula in Θ_j^2 sits on a clear identifiable node x in a tree in Δ_j^*, we have $\Delta_j^* = \Theta_j^{1*} + \Theta_j^{2*}$ and can therefore split Δ_j^* into $\tau_j^1 + \tau_j^2$, where $\tau_j^1 = \Theta_j^{1*}$ and $\tau_j^2 = \Theta_j^{2*}$, using x, with α in τ_j^2 and since $\Delta_j^* \vDash \delta_j^*$ we assume we can arrange things so that $\tau_j^1 \vDash \tau_j^2 + (\delta_1^j, \ldots, \delta_{t_j}^j)^* \to \delta$.

  We now define the case m. Assume $(\varphi_1, \ldots, \varphi_n) \vdash_m ((\psi_1, \ldots, \psi_s) \to \delta)$. We have in this case

  $$(\delta_1, \ldots, \delta_r) \to \delta, \Delta_1, \ldots, \Delta_{r-k}, \Delta_{r-k+1}(= \Theta_1 + \Theta_2), \Delta_{r-k+1}, \ldots, \Delta_r \vdash_m \delta$$

  with $(\Delta_1, \ldots, \Delta_{r-k}, \Theta_1) = (\varphi_1, \ldots, \varphi_n)$ and $(\Theta_2, \Delta_{r-k+2}, \ldots, \Delta_r) = (\psi_1, \ldots, \psi_s)$ and we have $\Delta_i \vdash_{m-1} \delta_i$, for $i \neq k+1$, and $\Delta_{r-k+1} = (\Theta_1 + \Theta_2) \vdash_{m-1} \delta_{r-k+1}$.

  We assume by induction that the following holds:

  $$\delta_j = (\delta_1^j, \ldots, \delta_{t_j}^j) \to q_j,\ j = 1, \ldots, r.$$

  Then, since $\Delta_j \vdash_{m-1} \delta_j$, we have $\Gamma_j = \Delta_j + \delta_1^j + \delta_{t_j}^j \vdash_{m-1} q_j$.

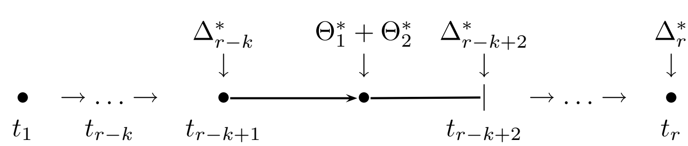


Figure 10.15: Standard translation

We assume that the database Γ_j^* is in standard form, of level $m-1$ and in particular for the case of $\Delta_{r-k+1} = \Theta_1 + \Theta_2$ we have $\Delta^*_{r-k+1} = \Theta_1^* + \Theta_2^*$ and that we have $\Theta_1^* \vDash \Theta_2^* \to \delta^*_{r-k+1}$ and Θ_1^* is ready for input into Θ_2^* to form $\Theta_1^* + \Theta_2^*$.

We now form the following level m database for

$$\varphi_1, \ldots, \varphi_n + \psi_1, \ldots, \psi_s \vdash_m \delta.$$

We start with the tree of Figure 10.14, where the δ_i^* in it are understood to be the c_i^* of our context here of case m. We consider the trees Δ_j^* and we

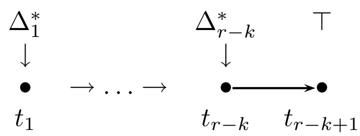


Figure 10.16: Tree τ' of the split

let the standard translation for case m be $\{\Delta_j^*\} = \tau$. We now need to show

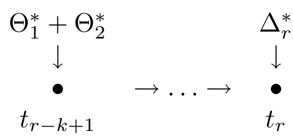


Figure 10.17: Tree τ'' of the split

the translations of $(\varphi_1, \ldots, \varphi_m)^*$ and ψ^*. Consider the node t_{r-k+1} in the standard translation. We have the situation in Figure 10.15. Our problem is that we want to split this tree in the middle of the subtree $(\Theta_1 + \Theta_2)^*$, with Θ_1^* going into $(\varphi_1, \ldots, \varphi_n)^*$ and Θ_2^* going into ψ^* along with Δ^*_{r-k+2} etc. Remember that Figure 10.15 is one big tree Δ^* which proves $\delta, (\Delta^* \vDash \delta)$. This tree can be split along two points. The first point is t_{r-k+1}. The two trees τ' and τ'' are in Figures 10.16 and 10.17.

We now split the tree in Figure 10.17 by taking out Θ_1^*. We get the two trees τ_3 and τ_2'' in Figures 10.18 and 10.19.

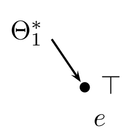


Figure 10.18: Tree τ_1'' of the split of τ''

We now define $(\varphi_1, \ldots, \varphi_n)^*$ and ψ^* as follows:

- $(\varphi_1^*, \ldots, \varphi_n)^* = \{\tau', \tau_1''\}$;
- $\psi^* = \tau_2'' \to \delta$.

It is clear that $(\varphi_1, \ldots, \varphi_n)^* \vDash \psi^*$ because if we input the trees of Figures 10.16 and 10.18 into the tree of Figure 10.19, we get the tree of Figure 10.15 which proves δ. ∇

Input points:
e in Θ_2^*
and t_{r-k+1}

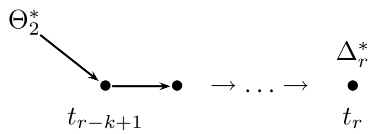


Figure 10.19: Tree τ_2'' of the split of τ_2

To illustrate the translation we provide in the sequel some examples.

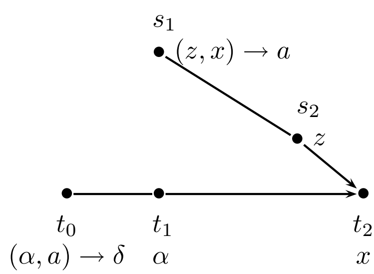


Figure 10.20: Tree τ_1

Example 10.2.17 Consider the formula

$$((z, x) \to a, z, x) \to a.$$

Observe that this formula is a theorem of concatenation logic. Consider now the following formula

$$(\alpha, a) \to q, \alpha, (z, x) \to a \vdash (z, x) \to \delta.$$

The translation of this formula is done as explained below.

First of all we translate the formula

$$(\alpha, a) \to \delta, \alpha, (z, x) \to a, z, x \vdash \delta$$

into the tree τ_1 in Figure 10.20.

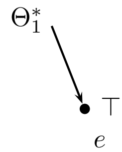


Figure 10.21: Tree τ_2

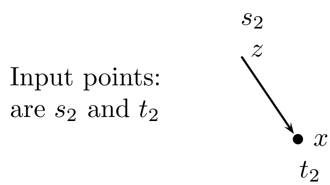


Figure 10.22: Tree τ_3

Then we split the tree at points t_2 and s_2 and get the trees τ_2, τ_3 and τ_4 in Figures 10.21, 10.22 and 10.23.

We have that the formula

$$((\alpha, a) \to q, \alpha, (z, x) \to a)$$

is translated into $\{\tau_4, \tau_4\}$. Moreover, the formula

$$(a, x) \to \delta$$

is translated to $\tau_2 \to \delta$.

We have $\{\tau_4, \tau_4\} \vDash \tau_3 \to \delta$ because $\{\tau_4, \tau_5\} + \tau_3 = \tau_1 \vDash \delta$, where τ_5 is the tree in Figure 10.24. $\triangledown$

Example 10.2.18 Consider now

$$((\alpha, a) \to q, \alpha, (z, x) \to a) \to b, (\alpha, a) \to q, \alpha, (z, x) \to a \vdash b.$$

In this case the translation is straightforward. It is the sequence of formulas as is. Thus, the part $((\alpha, a) \to q, \alpha, (z, x) \to a)$ is translated differently in this example as compared with Example 10.2.17. This is because we translate proofs, and not databases. In fact, we cannot translate this sequence on its own. $\triangledown$

Remark 10.2.19 Note that there are several types of translations.

1. *Uniform translation*
   Concatenation logic is complete for semigroup semantics with left identity (models of the form $(S, \circ, 1)$.)

   A uniform translation into classical logic would be as follows:

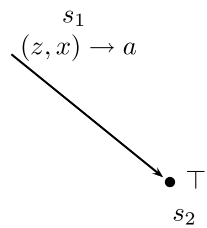


Figure 10.23: Tree τ_4

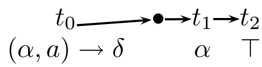


Figure 10.24: Tree τ_5

(i) $[q_i](x) = Q_i(x)$, for q_i atomic and $Q_i(x)$ a unary predicate of classical logic with a free variable x.

(ii) $[\varphi \to \psi](x) = \forall y([\varphi](y) \Rightarrow [\psi](x \circ y))$.

In the concatenation logic we have $\vdash \varphi$ if and only if in the theory of semigroup with left identity $\vdash [\varphi](1)$.

2. *Local translation*
This translation can translate any φ into an φ^* but such translation cannot be done uniformly. An example of that is translating from Basic into Prolog. Being both Turing machines, any Basic program can be translated (or associated with) a Prolog program doing the same job but probably not uniformly line by line.

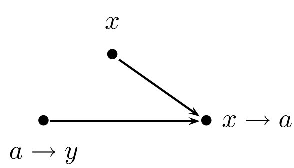


Figure 10.25: Database not proving y

3. *Proof translation*
This is the translation we offer in this section. If φ is a theorem of concatenation logic, it has a unique proof. We use this proof to translate. We cannot translate non-theorems. To see what is going on, let us try and translate from linear logic. Consider again $a \to y, x, x \to a \vdash y$. This is a valid statement of linear logic because of commutativity $\varphi \to (\psi \to \delta)$ is equivalent to

$\psi \to (\varphi \to \delta)$. We cannot translate $\mathbf{K}_{\text{past}}$ in that order because the database in Figure 10.25 does not prove y. ∇

We need to prove the statement that for any φ there exists an equivalent φ' in linear logic such that in linear logic $\vdash \varphi'$ if and only if in concatenation logic $\vdash \varphi'$ and then we translate φ' or we change the data-structure of $\mathbf{K}_{\text{past}}$ a little.

10.3 Input output networks

Logical input/output operations are generated, in the propositional case, by a set Δ of pairs of Boolean formulas (α,β). When an input formula is presented it is imaged by Δ, with the help of the classical consequence operation C.

The simplest way in which this is done is by putting $out_\Delta(\alpha) = \mathsf{C}(\Delta(\mathsf{C}(\alpha))$, where $\Delta(S)$ is just the image under Δ (considered as a relation) of the set S of formulas. This is known as the simple-minded input/output operation. Stronger operations that allow a more sophisticated treatment of disjunctive input, recycling of output as input, and/or automatic acceptance of inputs as outputs, are defined by adding to this simple definition. See [188, 189] for details.

We start with two observations. First, we can generalize the notion of an individual logical input/output operation in certain ways. In particular:

- we may wish to accept as inputs and outputs items that are more complex than bare propositions, carrying additional structure or other information;

- instead of using classical consequence to prepare input and package output as in the above definition, we might use other operations or relations of logic or information processing.

The input/output operations that we will consider will each be generalized in such ways. Nevertheless, they will have as a common core the composition of three operations. Given as *input* some propositional object α (perhaps a proposition, perhaps a more complex propositional structure, with or without labels), we apply a *familiar logical operation* (perhaps classical consequence, perhaps something else) to it, then take the *image* of that under a given set Δ of *generators* (pairs of propositional objects), and finally apply to that another *familiar logical operation* (not necessarily the same as the first).

The second observation is that it is also possible to combine input/output operations into networks, with the output of one operation serving as input to others. Here a great many possibilities suggest themselves. For example, the nodes in the net need not all correspond to the same kind of input/output operation. For instance, some may be simple-minded, others reusable, others disjunctive. Indeed, a single node may be labeled with several input/output operations, with the decision which to apply depending in part on the information coming in to it.

The way in which a node processes its inputs may depend on whether they are fresh or repeated, as well as on their propositional content. The paths leaving a node to others may be fixed, or may be activated/deactivated or even created/destroyed according to the information in the node. Moreover, nodes themselves may be created according to the information currently in a given node. Recursion is allowed, in that nodes may be labeled not only with individual input/output operations, but also with entire nets.

We will not attempt to define formally the family of all possible input/output logic nets, but leave the concept open-ended for expansion as needed to deal with examples. Nevertheless, each such net will have as its core a set of nodes, each with a label that contains an input/output operation and possibly more items. Information will come into the net through a fixed entry node, and after processing will leave through a fixed exit node.

The nets we consider are always finite, with deterministic behavior. We assume that the distribution of information may be analyzed by stages. That is, if we wish to determine the output of a specific net with a given input, we assume that there is a succession of net-states:

- state 0 represents the situation when the entry node d receives the input;
- state n is followed by a net-state $n{+}1$, uniquely determined by state n.

Of course, the state of a node in the net is not in general determined by its own immediately prior state, but on that of the entire net.

In each example it will turn out that after a given input, the state of the net and, in particular, the state of its exit node eventually stabilizes. This, however, needs to be proved in each case. This is the general conceptual framework. We now proceed to present some case studies: resolution in Prolog and one form of non-monotonic inference.

Example 10.3.1 As a case study we analyze Prolog as a network of input output logics. A Prolog clause has the form

$$C_i : \bigwedge A^i_j \to Q^i$$

where Q^i is the head and A^i_j are members of the body. We write the clause as

$$(Q^i, (A^1_1, \ldots, A^1_{k_i})).$$

If A^i_j does not exist we write $(Q^i, \emptyset)$.

To turn a clause into an input output node we consider Q^i as the trigger and $(A^i_1, \ldots, A^i_{k_i})$ as the output. An input to such a node is a sequence $(Q_1, \ldots, Q_m)$. Put the clauses as a network as in Figure 10.26. This is a circular network, with $\mathbf{d} = \mathbf{e} = C_1$ and $C_1RC_2, \ldots, C_{n-1}RC_n, C_nRC_1$.

When $(Q_1, \ldots, Q_m)$ is input into C_i we compare Q_1 and Q^i as follows:

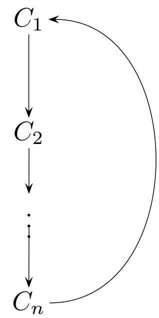


Figure 10.26: Circular network

- if they are different atoms then we output $(Q_1, \ldots, Q_n)$ to C_{i+1} (or C_1 if $i = n$);
- if the input is $\emptyset$ we pass it on as output as if $Q^i \neq Q_1$;
- if $Q_1 = Q^i$ we output $(A_1^i, \ldots, A_{k_i}^i Q_2, \ldots, Q_n)$ into C_{i+1}.

Note that if the clause is $C_i = Q^i$ then the output is $(Q_2, \ldots, Q_m)$. Suppose we give some single input into C_1 and let it run. The network evolves onto a steady state of either $\emptyset$ as repeated input or some non-empty input $(Q_1, \ldots, Q_{m'})$. The first is success, the second is failure.

Example 10.3.2 We now discuss the case of non-monotonic logic. The non-monotonic consequence $\mathrel{|\!\sim}$ is defined as the smallest consequence containing Γ, closed under $\vdash$ and the rules:

1. $\Delta, \mathrel{|\!\sim} A$
2. $\dfrac{\Delta, A \mathrel{|\!\sim} B; \Delta \mathrel{|\!\sim}}{\Delta \mathrel{|\!\sim} B}$
3. $\dfrac{\Delta \mathrel{|\!\sim} A; \Delta \mathrel{|\!\sim} B}{\Delta, A \mathrel{|\!\sim} B}$

A more constructive way of generating the consequences of a theory Δ is to proceed as follows. A set of assumptions Δ is said to be *reducible* to $\Delta' \subseteq \Delta$ if for some $A_1, \ldots, A_k$ we have

$$\Delta = \Delta' \cup \{A_1, \ldots, A_k\} \text{ and } \Delta' \mathrel{|\!\sim} A_i, \; i = 1, \ldots, k.$$

We use the word "reducible" because by a property of $\mathrel{|\!\sim}$ we should look at all Y such that $\Delta' \mathrel{|\!\sim} Y$ and that would imply $\Delta \mathrel{|\!\sim} Y$. So, to generate the set of all Y such that $\Delta \mathrel{|\!\sim} Y$, we should look at all irreducible subsets $\Delta' \subseteq \Delta$ to which Δ is reducible (all these subsets Δ-pools) and generate all their consequences.

It is therefore sufficient for the purpose of showing how input output networks can generate non-monotonic consequence to assume that we are dealing with irreducible theories. For such cases, the proof theory for non-monotonic logic goes as follows (see [107]):

- we start with a set of pairs $\{(\Gamma_i, \beta_i)\}$ and a monotonic logic base $\vdash$;
- we consider a theory Δ;
- we look at all pairs (Γ, β) such that $\Delta = \Gamma$;
- finally, we non-monotonically conclude β and write $\Delta \mid\!\sim \beta$.

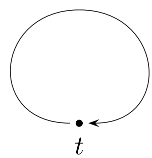


Figure 10.27: Net

We get a set Θ_0 of outputs $\beta_1^0, \beta_2^0, \ldots$. We now continue to accumulate consequences by feeding in as inputs any set

$$\Delta' = \Delta \cup \{\beta_1, \ldots, \beta_k, \beta_i \in \Theta_0\}$$

(if $k = 0$ then $\Delta' = \Delta$). This triggers a new set $\Theta_1 = \{\beta_1^1, \beta_2^1, \ldots\}$.

We have $\Theta_0 \subseteq \Theta_1$ because one option is to feed Δ itself. We now carry on inductively and feed

$$\Delta' = \Delta \cup \{\beta_1, \ldots, \beta_k : \beta_i \in \Theta_m\}$$

and get the set Θ_{m+1}. Notice that Δ always appears in the conjunction, because we are dealing with non-monotonic logic. This process corresponds to the net of Figure 10.27. ∇

The basic idea of input output logic is that given an initial data Δ and an input formula A an output formula B is affected. This schema is immediately reminiscent of substructural implication where

$$\Delta \vDash A \Rightarrow B \text{ if and only if } \Delta \circ A \vDash B.$$

This suggests semigroup modeling of input output logic. We shall see, however, that more general models (LDS models) are required, if we want to have networks of such logics. The basic nodes of these networks are labels. If a node t is connected

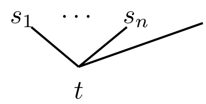


Figure 10.28: Nodes in a network

to nodes $s_1, \ldots, s_n$, see Figure 10.28, and t receives an input X, the output from t goes on to become input for $s_1, \ldots, s_n$.

Thus, we imagine an input–output node t as comprising two metaformulas: $\varphi_t(X)$ and $\gamma_t(X)$. The input X is given to t and the output is $\varphi_t(X)$. However, the node t may not be able to process X completely. It therefore *creates* (or activates) a new node $s = g(t)$ and gives it $\gamma_t(X)$ as input. Furthermore, for certain X, the node may not process at all and X remains in the node.

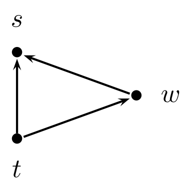


Figure 10.29: From t input X can go to w or to s

The node $s = g(t)$ that t creates with input $\gamma_t(X)$ is assumed, for simplicity, to be a copy of t itself, that is, we have $\varphi_s = \varphi_t$ and $\gamma_t = \gamma_s$. We assume further that $\gamma_t(X)$ satisfies the following well foundedness condition: for every X there is a natural number n such that:

$$\gamma_t^n(X) = \top.$$

We also assume

$$\varphi_t(\top) = \top \quad \text{and} \quad \gamma_t(\top) = \emptyset$$

for all t. The new node $s = g(t)$ that t creates is connected in the network to some of the nodes to which t is connected (for instance, all or none are possible simple options).

It is also possible to have hypernetworks where the nodes connected depend on the processing history of the input. In Figure 10.29, from t, X can go to w and then to s or directly to s. From s, where the output goes depends on its history.

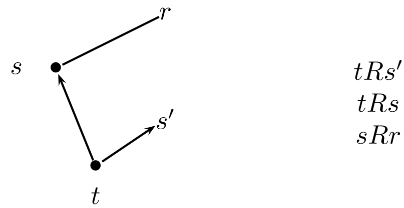


Figure 10.30: Example of a network

Example 10.3.3 We now consider modal logic as input output logic:

Consider a network as in Figure 10.30. All nodes have the same metaformulas φ and γ:

- $\varphi(\Box X)$ is X;
- $\gamma(\Diamond X)$ is to create a new node s, connected to t only, and input X to it.

Let the input be $(\Box((\Box X) \wedge (\Diamond Y)))$. Then:

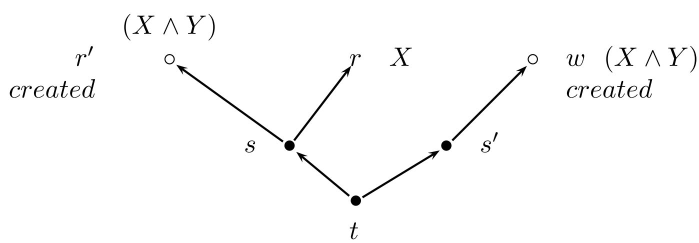


Figure 10.31: Inputing $(\Box((\Box X) \wedge (\Diamond Y)))$ to t

- t outputs $((\Box X) \wedge (\Diamond Y))$ to s and to s';
- s outputs X to node r and also creates a new node r' and outputs to it $(X \wedge Y)$;
- sRr' holds.

Similarly, s' creates w with $s'Rw$ and outputs to it $(X \wedge Y)$. Figure 10.31 shows what we get now after input. Note that in modal logic all processors do the same input output. $\triangledown$

The simplest variation is to change the meaning of $\Diamond$ into $\Diamond\!\!\!\!-$.

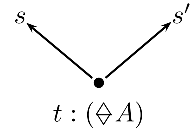


Figure 10.32: Input of $(\diamondminus A)$

Whereas $(\Diamond A)$ creates a node unconnected to any other, $(\diamondminus A)$ creates a world connected to the current worlds or to all the accessible worlds.

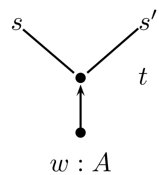


Figure 10.33: Creation of a new w as a result of inputing $(\diamondminus A)$

Option 1
The input of $\diamondminus A$ in Figure 10.32 creates a node w as in Figure 10.33. This makes $\diamondminus$ mean "yesterday". Otherwise, we can open a parallel unit as in Figure 10.34. The major difference between input output nets and modal logic is that the output is different at different nodes. Let us go to the original network of Figure 10.30. Let us input X at t. We get, in general, the situation of Figure 10.35. Let $\mathcal{A}$ be

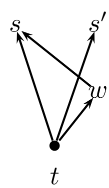


Figure 10.34: Parallel unit with $tRw \wedge wRs \wedge wRs'$

a set of labels and let R be a binary relation on labels. Let $t \mapsto (\varphi_t, \gamma_t)$ be input output functions associated with labels. Let x, y, z be variable labels and let τ be a theory on labels constituted by Horn clauses, as, for instance,

$$\tau = \forall xyz(xRz \wedge xR^5 y \to zRy).$$

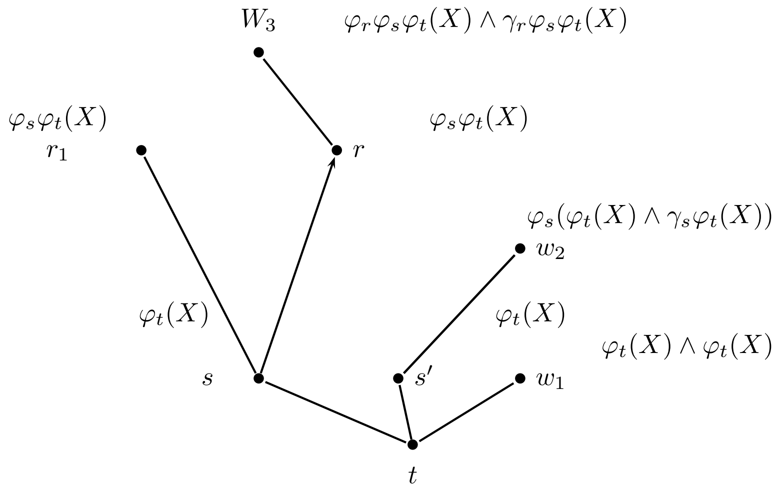


Figure 10.35: Inputing X at t

Then, τ is a closure condition to be used when we create points. For example, suppose we have the situation of Figure 10.36. Then τ forces us to join z to x_5.

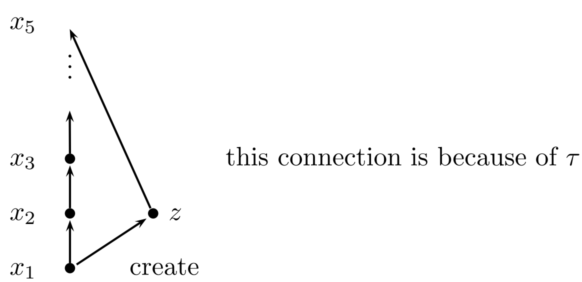


Figure 10.36: Joining node z

Definition 10.3.4 A *network* is a tuple

$$N = \langle T, R, \mathbf{d}, \mathbf{e} \rangle$$

where T is a set (the set of nodes, or points), $R \subset T \times T$, $\mathbf{d}$ is the input point and $\mathbf{e}$ is the output point. We also consider the functions $\lambda t.\varphi_t$, $\lambda t.\gamma_t$ and τ.

We say that $X \vdash_\Delta Y$ if with input X into $\mathbf{d}$ we get Y at $\mathbf{e}$. To do this we need proof theory. It is better to assume that input comes to every node. So the database Δ is a network with input A_i already at every node t_i. Output is any B such that $\Delta \vdash \mathbf{e} : B$.

The *proof rules* are as follows:

$$\textit{export} \qquad \frac{t : A \quad tRs}{s : \varphi_t(A)}$$

$$\textit{creation} \qquad \frac{t : A}{\text{create } w, \bigwedge_i wRt_i \quad w : \varphi_t(A) \wedge \gamma_t(A)}$$

$$\textit{closure} \qquad \frac{\tau = \bigwedge x_i R y_i \to tRs, x_i R y_i}{tRs}$$

∇

We now present an example that illustrates the application of the rules above.

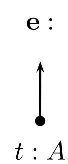


Figure 10.37: Database Δ

Example 10.3.5 Consider the following situation:

- $\tau = (xRy \wedge xRz) \to yRz$;
- $\varphi_E(A) \circ E$, where "$\circ$" denotes AGM revision (see [4]);
- $\gamma_E(A)$ = create a possible node with A.

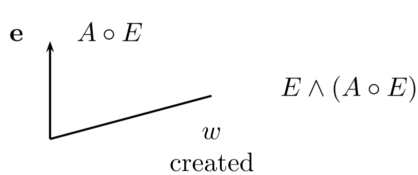


Figure 10.38: Database after applying the rules to Δ

Let Δ be as in Figure 10.37. We use the rules and get Figure 10.38.

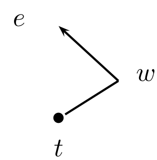


Figure 10.39: Database after applying the closure rule

Then, we have

$$E \wedge (A \circ E) = A \circ E.$$

Using now the closure rule we get the situation depicted in Figure 10.39. Finally, we use rule on w and get Figure 10.40.

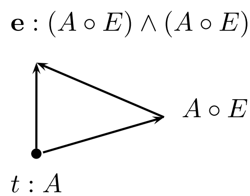


Figure 10.40: Database after applying rule to w

$\triangledown$

10.4 Fibring neural networks

The goal of neural-symbolic integration is to benefit from the symbolic and the connectionist paradigms of artificial intelligence [56, 78]. Towards this end, efficient, parallel and distributed learning capability should be at the core of any neural-symbolic system and, one may argue, of any artificial intelligence system.

A neural network consists of interconnected neurons (or processing units) that compute a simple function according to the weights (real numbers) associated to the connections. Learning in this setting is the incremental adaptation of the weights [143]. The interesting characteristics of neural networks do not arise from the functionality of each neuron, but from their collective behaviour. For more details about neural networks see, for instance, [143].

Neural-symbolic systems that use simple neural networks, such as single hidden layer feedforward or recurrent networks, typically only manage to represent and reason about propositional symbolic knowledge or *if then else* rules [28, 78, 100, 220, 261]. On the other hand, neural-symbolic systems that are capable of representing and of reasoning about more expressive symbolic knowledge, such as modal logic and first-order logic, normally are less capable of learning new concepts efficiently [150, 253, 247, 161].

It is important to strike a balance between the reasoning and learning capabilities of neural-symbolic systems. Either the simple networks to which, for example, the efficient Backpropagation learning algorithm, or its variations, can be applied to [230, 271, 272] must be shown to represent languages more expressive than propositional logic, or the complex connectionist systems that are capable of representing first order logic, such as for example CHCL [151] must have efficient learning algorithms developed for them. This is so because real-world applications (such as failure diagnosis, engineering and bioinformatics applications) do require the use of languages more expressive than propositional logic. Bioinformatics, in particular, very much depends on the ability to represent and reason about relations as used in first order logic [6].

In this case study we extend simple networks that use Backpropagation in order to allow for higher expressive power. We do so by following the fibring by functions mechanism [108], already referred to in Subsection 1.2.3 of Chapter 1 (see also Section 3.4 of Chapter 3). To this end, we know that a fundamental aspect of symbolic computation lies on the ability to do recursion. As a result, to make neural networks behave like logic, we need to add recursion to it by allowing networks to be composed not only of interconnected neurons but also of other networks.

Figure 10.41 exemplifies how a network can be embedded into another. Of course, the idea of fibring is not only to organize networks as a number of sub-networks. In Figure 10.41, for example, the hidden neuron of network A is expected to be a neural network (network B) in its own right, and the input, weights and output of network B may depend on the activation values of neurons in network A, according to the fibring function used. For example, a fibring function may be to multiply the weights of network B by the input potential of network's A output neuron.

Most of the work on how to do recursion in neural networks has concentrated on the use of recurrent auto-associative networks and symmetric networks to represent formal grammars [87, 260, 248, 249, 222]. In general, the network learns how to simulate a number of recursive rules by similarity, and the question of how such rules are represented in the network is treated as secondary. In this case study, we present a different treatment to the subject, looking at it from a neural-symbolic integration perspective [78]. The idea is to be able to represent and learn symbolic rules of the form

$$a \to (b \to c)$$

where $(b \to c)$ would be encoded into network B and then $a \to (b \to c)$ would be encoded into the fibred network containing networks A and B .

We introduce and define below the fibred neural network (fNN) architecture, and show that, in addition to being universal approximators, fNNs can approximate any polynomial function, thus being more expressive than standard feedforward networks. Briefly, this can be shown by noting that fibred neural networks compute, for instance, the function $f(x) = x^2$ exactly for any given input x in $\mathbb{R}$, as opposed to feedforward networks which are restricted to compact (that is, closed and bounded) domains [70, 152].

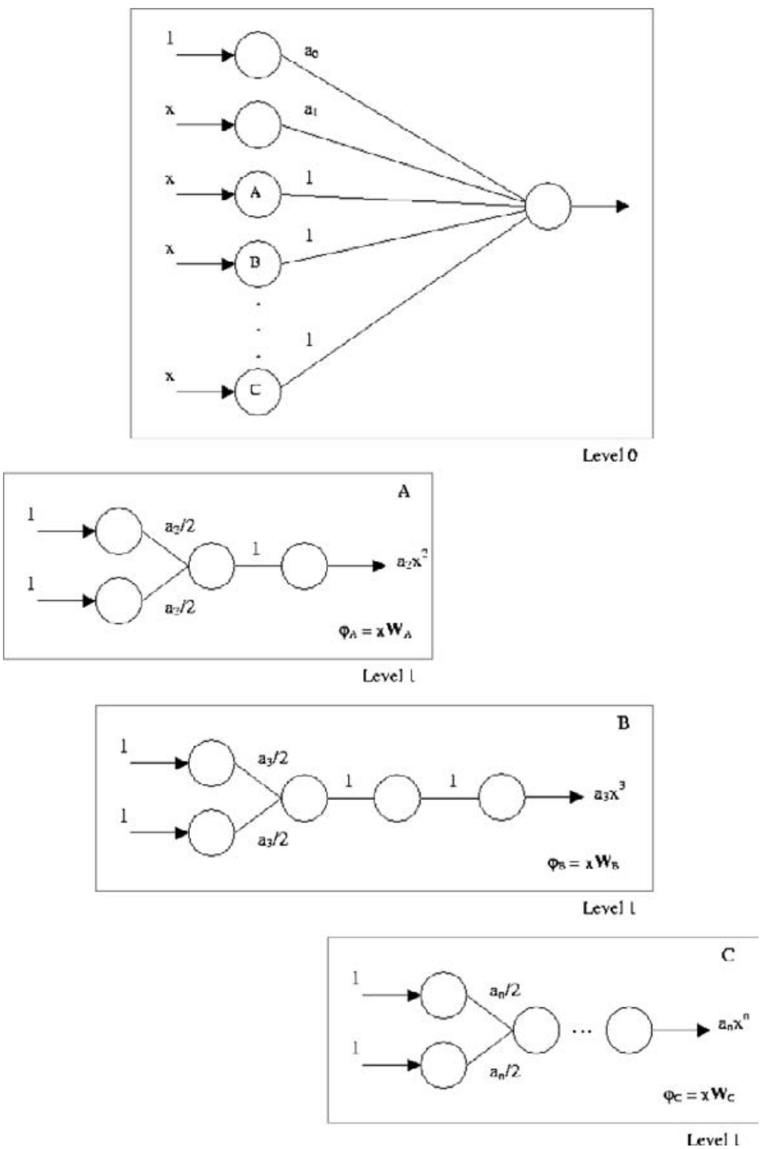


Figure 10.41: Fibring neural networks

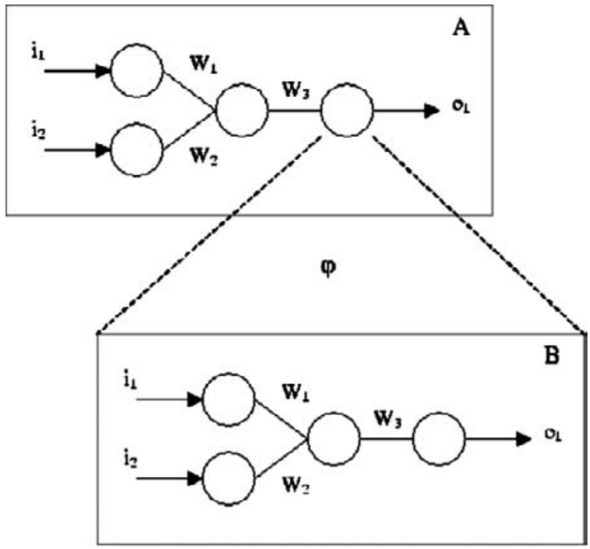


Figure 10.42: Fibring two simple networks

Intuitively, fibring neural networks can be seen as running and training neural networks at the same time. In Figure 10.41, for example, at the same time that we run network A, we perform a kind of learning in network B because we allow the weights of B to change according to the fibring function. In other words, object-level network running and meta-level network training are occurring simultaneously in the same system, and this is responsible for the added expressiveness of the system.

The main idea behind fibring neural networks is to allow single neurons to behave like entire embedded networks according to a fibring function φ. This function qualifies the function computed by the embedded network so that the embedded network's output depends on φ. For example, consider network A and its embedded network (network B) in Figure 10.41.

Let $\mathbf{W}_A$ and $\mathbf{W}_B$ be the set of weights of network A and network B respectively. Let

- $f_{\mathbf{W}_A}(\mathbf{i}_A)$ be the function computed by network A;
- $g_{\mathbf{W}_B}(\mathbf{i}_B)$ be the function computed by network B;

where $\mathbf{i}_A$ and $\mathbf{i}_B$ are the input vectors of networks A and B respectively. If network B is embedded into network A with fibring function φ, the function computed by network B becomes $g_{\varphi(\mathbf{W}_B)}(\mathbf{i}_B)$, and then the function computed by network A becomes $f_{\mathbf{W}_A, g_{\varphi(\mathbf{W}_B)}(\mathbf{i}_B)}(\mathbf{i}_A)$, as the following example illustrates.

Consider the two simple networks (A and B) of Figure 10.42. Let us assume, without loss of generality, that input and output neurons have the identity as activation function, while hidden neurons have $h(x) = \tanh(x)$ as activation function [152]. We use bipolar inputs $i_j \in \{-1, 1\}$, $W_{jk} \in \mathbb{R}$, and outputs $o_k \in \{-1, 1\}$.

The output of network A is

$$o_1^A = W_3^A.h(W_1^A i_1^A + W_2^A i_2^A)$$

and the output of network B is

$$o_1^B = W_3^B.h(W_1^B i_1^B + W_2^B i_2^B).$$

Now, let network B be embedded into network A as shown in Figure 10.42. This indicates that the input potential of A's output neuron will influence B according to fibring function φ. Let us refer to the input potential of A's output neuron as $\mathbf{I}(o_1^A)$. In addition, this indicates that the output of B (o_1^B) will influence A (in this example, only the output of A). Note that, in this particular example, $\mathbf{I}(o_1^A) = o_1^A$ due to the use of the identity as activation function in the output layer.

Suppose $\varphi(\mathbf{W}_B) = \mathbf{I}(o_1^A) \cdot \mathbf{W}_B$, where $\mathbf{W}_B = [W_1^B, W_2^B, W_3^{\mathbf{B}}]$. Let us use $\overline{o}_1^A$ and $\overline{o}_1^B$ to denote the outputs of networks A and B respectively, after they are fibred. Then $\overline{o}_1^B$ is obtained by applying φ to $\mathbf{W}_B$ and calculating the output of such a network, as follows:

$$\overline{o}_1^B = (\mathbf{I}(o_1^A).W_3^B) \cdot h((\mathbf{I}(o_1^A).W_1^B)i_1^B + (\mathbf{I}(o_1^A).W_2^B)i_2^B).$$

Moreover, $\overline{o}_1^A$ is obtained by taking $\overline{o}_1^B$ as the output of the neuron in which network B is embedded. In this example,

$$\overline{o}_1^A = \overline{o}_1^B.$$

Notice how network B is being trained (when φ changes its weights) at the same time that network A is running.

Clearly, fibred networks can be trained from examples in the same way that standard feedforward networks are (for example, with the use of Backpropagation [230]). Networks A and B of Figure 10.42, for example, could have been trained separately before being fibred. Network A could have been trained, for instance, with a robot's visual system, while network B would have been trained with its planning system. For simplicity, we assume for now on that, once defined, the fibring function itself should remain unchanged.

In addition to using different fibring functions, networks can be fibred in a number of different ways as far as their architectures are concerned. The networks of Figure 10.42, for example, could have been fibred by embedding network B into an input neuron of network A(say, the one with input i_1).

In this case, outputs $\overline{o}_1^B$ and $\overline{o}_1^A$ would have been

$$\overline{o}_1^B = \varphi(W_3^B) \cdot h(\varphi(W_1^B)i_1^B + \varphi(W_2^B)i_2^B)$$

where φ is a function of $\mathbf{W}_B$ (say, for instance, $\varphi(\mathbf{W}_B) = i_1 \cdot \mathbf{W}_B$), and

$$\overline{o}_1^A = W_3^A.h(W_1^A \overline{o}_1^B + W_2^A i_2^A).$$

Let us now consider an even simpler example that, nevertheless, illustrates the power of fibring neural networks. Consider two networks A and B , both with a single input neuron (i^A and i^B, respectively), a single hidden neuron and a single output neuron (o^A and o^B, respectively). Let all the weights in both networks have value 1, and let the identity f be the activation function of all the neurons (including the hidden neurons). As a result, we simply have

- $o^A = f(W_2^A \cdot f(W_1^A \cdot f(i^A))) = i^A$;
- $o^B = f(W_2^B \cdot f(W_1^B \cdot f(i^B))) = i^B$;

where W_1^A and W_2^A are the weights of network A, and W_1^B and W_2^B are the weights of network B . Now, if we embed network B into the input neuron of network A, we obtain

- $\overline{o}^B = f(\varphi(W_2^B) \cdot f(\varphi(W_1^B) \cdot f(i^B)))$;
- $\overline{o}^A = f(W_2^A \cdot f(W_1^A \cdot \overline{o}^B))$.

Since f is the identity, we have

- $\overline{o}^B = \varphi(W_2^B) \cdot \varphi(W_1^B) \cdot i^B$;
- $\overline{o}^A = W_2^A \cdot W_1^A \cdot \overline{o}^B$.

Now, let the fibring function be

$$\varphi(\mathbf{W}_A, \mathbf{i}_A, \mathbf{W}_B) = i^A \cdot \mathbf{W}_B$$

where $\mathbf{W}_B = [W_1^B, W_2^B]$. Since $W_1^A, W_2^A, W_1^B, W_2^B$ are all equal to 1, we obtain $\overline{o}^B = i^A \cdot i^A \cdot i^B$ and $\overline{o}^A = \overline{o}^B$. This means that if we fix $i^B = 1$, the output of network A (fibred with network B) will be $i^A \cdot i^A$.

Finally, assume that the following sequence is given as input to A fibred with B :

$$n, 1/n, n+1, 1/(n+1), n+2, 1/(n+2), \ldots$$

for $n \in \mathbb{R}$. The corresponding output sequence of A will be:

$$n^2, 1, (n+1)^2, 1, (n+2)^2, 1, \ldots$$

Note that, input n changes the weights of B from 1 to n, input $1/n$ changes the weights of B back to 1, input $n+1$ changes the weights of B from 1 to $n+1$, input $1/(n+1)$ changes the weights of B back to 1, and so on. Note that, since the fibring function changes the weights of the embedded network, we use $1/n$, $1/n+1$, $1/n+2\ldots$ to *reset* the weights back to 1 in the sequence computation.

The interest in this sequence lies in the fact that, for alternating inputs, the square of the input is computed exactly by the network for any input in $\mathbb{R}$. This illustrates an important feature of fibred neural networks, namely, their ability to approximate functions in an unbounded domain. This results from the recursive characteristic of fibred networks as indicated by the function $f_{\mathbf{W}_1, g_{\mathbf{W}_2}(\mathbf{i}_2)}(\mathbf{i}_1)$ computed by the network, and will be discussed in more detail below.

Now we define fibred neural networks (fNNs) precisely, we define the dynamics of fNNs, and we show that fNNs can approximate unbounded functions.

For the sake of simplicity, we restrict the definition of fibred networks to feedforward, single output neuron networks. We also concentrate on networks with linear input and output activation functions, and either linear or sigmoid hidden layer activation function. We believe, however, that the principles of fibring could be applied to any artificial neural network model. In what follows, we allow not only two networks, but any number of embedded networks to be nested into a fibred network. We also allow for an unlimited number of hidden layers per network.

Definition 10.4.1 Let A and B be two neural networks. A function $\varphi_n : \mathbf{I} \rightarrow \mathbf{W}$ is called a *fibring function* from A to B if $\mathbf{I}$ is the input potential of a neuron n in A and $\mathbf{W}$ is the set of weights of B. $\triangledown$

Definition 10.4.2 Let A and B be two neural networks. We say that B is *embedded into* A if φ_n is a fibring function from A to B and the output of neuron n in A is given by the output of network B. The resulting network composed of networks A and B is called a *fibred neural network*. $\triangledown$

Note that many networks can be embedded into a single network, and that networks can be nested so that network B is embedded into network A, network C is embedded into network B, and so on. The resulting fibred network can be constructed by applying Definition 10.4.2 recursively, that is, first to embed C into B and then to embed the resulting network into A.

Example 10.4.3 Consider three identical network architectures (A, B and C), each containing a single linear input neuron, a single linear hidden neuron, and a single linear output neuron. Let us denote the weight from the input neuron to the hidden neuron of network $x \in \{A, B, C\}$ by W_x^h, and the weight from the hidden neuron to the output neuron of x by W_x^o.

Assume we embed network C into the output neuron of network B, and embed the resulting network into the output neuron of network A (according to Definition 10.4.2), as depicted in Figure 10.43. Let

- φ_B denote the fibring function from A to B and define $\varphi_B = i_A^o \cdot \mathbf{W}_B$;
- φ_C denote the fibring function from B to C and define $\varphi_C = i_B^o \cdot \mathbf{W}_C$;

where

- i_A^o is the input potential of A's output neuron given input x;
- i_B^o is the input potential of B's output neuron given inputs x and y;

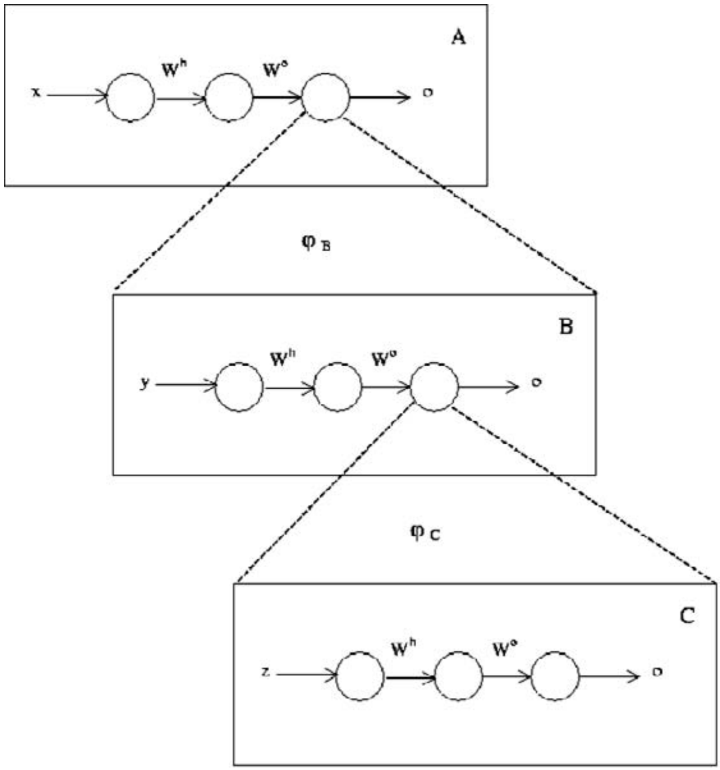


Figure 10.43: Nesting fibred networks

- $\mathbf{W}_B$ denotes the weight vector $[W_B^h, W_B^o]$ of B;
- $\mathbf{W}_C$ denotes the weight vector $[W_C^h, W_C^o]$ of C.

Initially, let $W_A^h = \sqrt{a}$, where $a \in \mathbb{R}^+$, and $W_A^o = W_B^h = W_B^o = W_C^h = W_C^o = 1$. As a result, given input x to A, the input potential of A's output neuron will be $x\sqrt{a}$. Then, φ_B will be used to update the weights of network B to $W_B^h = x\sqrt{a}$ and $W_B^o = x\sqrt{a}$. If we had only networks A and B fibred, input $y = 1$, for example, would then produce an output $o = ax^2$ for network B and then A.

Since network C is also embedded into the system, given input y, fibring function φ_C will be used to update the weights of network C, according to the input potential of B's output neuron.

Thus, given $y = 1$, the input potential of B's output neuron will be ax^2, and the weights of network C will change to $W_C^h = ax^2$ and $W_C^o = ax^2$. Finally, assume $z = 1$. The output o of networks C, B and A will be a^2x^4. This illustrates the computation of polynomials in fNNs. The computation of odd degree polynomials and of negative coefficients could be achieved with the addition of more hidden layers to the networks, as we will see in the sequel. ▽

Example 10.4.3 also illustrates the dynamics of fibred networks. Let us now define such a dynamics precisely.

Definition 10.4.4 Let N_1, $N_2, \ldots, N_n$ be neural networks. N_1, $N_2, \ldots, N_n$ form a *nested fibred network* if N_i is embedded into a neuron of N_{i-1} with a fibring function φ_i for any $2 \leq i \leq n$. We say that $j-1$, $1 \leq j \leq n$, is the *level* of network N_j. ▽

Definition 10.4.5 Let N_1,$N_2, \ldots, N_n$ be a nested fibred network. Let φ_i be the fibring function from N_{i-1} to N_i for $2 \leq i \leq n$. Let $\mathbf{i}_j$ denote an input vector to network N_j, $\mathbf{W}_j$ the current weight vector of N_j, $\mathbf{I}_n(\mathbf{i}_j)$ the input potential of N_j's neuron n_j into which N_{j+1} is embedded given input vector $\mathbf{i}_j$, $\mathbf{O}_{n_j}$ the output of neuron n_j, and $f_{\mathbf{W}_j}(\mathbf{i}_j)$ the function computed by network N_j given $\mathbf{W}_j$ and $\mathbf{i}_j$ as in the standard way for feedforward networks. The output o_j of network N_j, $1 \leq j \leq n-1$, is defined recursively in terms of the output o_{j+1} of network N_{j+1}, as follows:

$$\mathbf{W}_{j+1} := \varphi_{j+1}(\mathbf{I}(\mathbf{i}_j), \mathbf{W}_{j+1}),\ 1 \leq j \leq n-1$$

$$o_n = f_{\mathbf{W}_n}(\mathbf{i}_n)$$

$$o_j = f_{\mathbf{W}_j}(\mathbf{i}_j, \mathbf{O}_{n_j} := o_{j+1})$$

where $f_{\mathbf{W}_j}(\mathbf{i}_j, \mathbf{O}_{n_j} := o_{j+1})$ denotes the function computed by N_j substituting the output of its neuron n_j by the output of network N_{j+1}. ▽

Now that fNNs have been defined, we proceed to show that, in addition to being universal approximators, fNNs can approximate any polynomial function, and thus are more expressive than standard feedforward neural networks.

Proposition 10.4.6 *Fibred neural networks can approximate any (Borel) measurable function in a compact domain to any desired degree of accuracy.*

Proof. This follows directly from the proof that single hidden layer feedforward neural networks are universal approximators [152], together with the observation that level zero networks are a generalization of single hidden layer feedforward networks. ◁

The proposition above states that fNNs are universal approximators. For the next result, recall that, differently from functions in a compact domain, polynomial functions are not bounded.

Proposition 10.4.7 *Fibred neural networks can approximate any polynomial function to any desired degree of accuracy.*

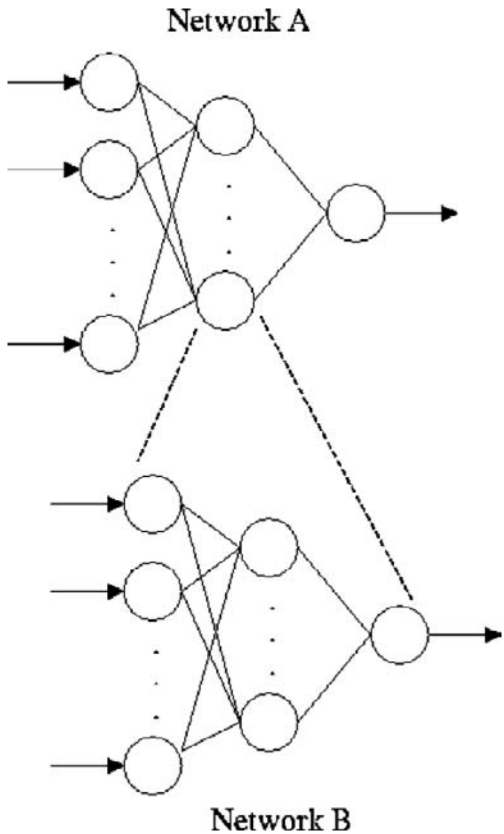


Figure 10.44: Computing polynomials in fibred networks

Proof. Consider the level zero network N of Figure 10.44, and its three embedded networks A, B and C at level 1, all containing linear neurons. Let $n+1$ ($n \in \mathbb{N}$) be the number of input neurons of N, $0 \leq i \leq n$, $a_i \in \mathbb{R}$.

We embed $n-1$ networks into the input neurons of N, each network representing $x^2, x^3, \ldots, x^n$, as indicated in Figure 10.44 for networks A, B and C, representing x^2, x^3 and x^n, respectively. A network N_j that represents x^j, $2 \leq j \leq n$, contains two input neurons (to allow the representation of $a_j \in \mathbb{R}$), $j-1$ hidden layers, each layer containing a single hidden neuron (let us number these $h_1, h_2, \ldots, h_{j-1}$), and a single output neuron. In addition, let $a_j/2$ be the weight from each input neuron to h_1, and let 1 be the weight of any other connection in N_j. We need to show that such a network computes $a_j x^j$.

From Definition 10.4.5, given input x to N and $\varphi_j = x\mathbf{W}_j$, the weights of N_j are multiplied by x. Then, given input $(1, 1)$ to N_j, neuron h_1 will produce output

$a_j x$, neuron h_2 will produce output $a_j x^2$, and so on. Neuron h_{j-1} will produce output $a_j x^{j-1}$, and the output neuron will produce $a_j x^j$.

Finally, by Definition 10.4.2, the neuron in N into which N_j is embedded will present activation $a_j x^j$, and the output of N will be $\sum_j a_j x^j$. The addition of $a_1 x$ and a_0 is straightforward (see Figure 10.44), completing the proof that fNNs compute $\sum_i a_i x^i$. ◁

10.5 Fibring Bayesian networks

Causal relations can themselves take part in causal relations. The fact that smoking causes cancer (SC), for instance, causes government to restrict tobacco advertising (A), which helps prevent smoking (S), which in turn helps prevent cancer (C). This causal chain is depicted in Figure 10.45.

Hence, causal models need to be able to treat causal relationships as causes and effects. This observation motivates an extension of the Bayesian network causal calculus to allow nodes that themselves take Bayesian networks as values. Such networks will be called *recursive Bayesian networks*.

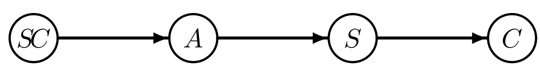


Figure 10.45: Causal relation

Because recursive Bayesian networks make causal and probabilistic claims at different levels of their recursive structure, there is a danger that the network might contradict itself. Hence, we need to ensure that the network is consistent. Having done this, we present a Markov condition which applies to recursive Bayesian networks, and which allows joint distributions over the domain to be determined by such networks. We also refer to other generalizations of Bayesian networks, and we show by analogy with recursive Bayesian networks how recursive causality can be modeled in structural equation models.

A recursive Bayesian network is an instance of a very general structure called a self-fibring network, whose properties are discussed in Section 10.6.

Bayesian networks

It is almost universally accepted that causality is an asymmetric binary relation (not quite universally, [205] disagrees for example). But the question of what the causal relation relates is much more controversial: the relata of causality have variously taken to be single-case events, properties, propositions, facts, sentences and more.

In this section we deal with cases in which causal relations themselves are included as relata of causality. Our aim is to shed light on the processes of causal reasoning, especially formalizations of causal reasoning.

More generally, we shall consider sets of causal relations, represented by directed causal graphs such as that of Figure 10.45, as relata of causality. A single causal relationship is then represented by a causal graph consisting of two nodes referring to the relata and an arrow from cause to effect. If, as in Figure 10.45, a causal graph G contains a causal relation or causal graph as a value of a node, we shall call G a *recursive causal graph* and say that it represents *recursive causality*.

Figure 10.46: Causal chain

Policy decisions are often influenced by causal relations. As we have already seen, smoking causing cancer itself causes restrictions on advertising. Similarly, monetary policy makers reduce interest rates (R) because interest rate reductions boost the economy (E) by causing borrowing increases (B) which in turn allow investment (I). Here we have a causal chain as in Figure 10.46 forming the node RE in Figure 10.47.

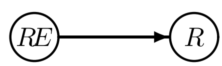


Figure 10.47: Node RE

Policy need not be made for us: we often decide how we behave on the basis of perceived causal relationships. It is plausible that drinking red wine causes an increase in anti-oxidants which in turn reduces cholesterol deposits, and this apparent causal relationship causes some people to increase their red wine consumption.

This example highlights two important points. Firstly, it is a belief in the causal relationship which directly causes the policy change, not the causal relationship itself. The belief in the causal relationship may itself be caused by the relationship, but it may not be: it may be a false belief or it may be true by accident. Likewise, if a causal relationship exists but no one believes that it exists, there will be no policy change. Secondly, the policy decision need not be rational on the basis of the actual causal relationship that causes the decision: drinking red wine may do more harm than good.

A contract can be thought of as a causal relationship, and the existence of a contract can be an important factor in making a decision. A contract in which production of commodity C is purchased at price P may be thought of as a causal relationship $C \longrightarrow P$, and the existence of this causal relationship can in turn cause the producer to invest in further means of production, or even other commodities.

For example, a Fair Trade chocolate company has a long-term contract with a co-operative of Ghanaian cocoa producers to purchase (P) cocoa (C) at a price

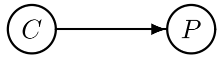


Figure 10.48: Causal relationship $C \to P$

advantageous to the producer as in Figure 10.48. The existence of this contract (CP) allows the cooperative to invest in community projects such as schools (S), as in Figure 10.49.

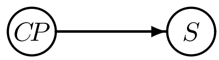


Figure 10.49: Causal relationship $CP \to S$

An insurance contract is an important instance of this example of recursive causality. Insuring a building against fire may be thought of as a causal relationship of the form "insurance contract causes [fire F causes remuneration R]" or, for short,

$$[C \longrightarrow P] \longrightarrow [F \longrightarrow R]$$

where, as before, C is the commodity (that is, the contract) and P is payment of the premium. The existence of such an insurance policy can cause the policy holder to commit arson (A) and set fire to her building and thereby get remunerated:

$$[[C \longrightarrow P] \longrightarrow [F \longrightarrow R]] \longrightarrow A \longrightarrow F \longrightarrow R.$$

Causality in this relationship is nested at three levels. Insurance companies will clearly want to limit the probability of remuneration given that arson has occurred.

Thus, we see that recursive causality is particularly pervasive in decision-making scenarios. However, recursive causality may occur in other situations too — situations in which it is the causal relationship itself, rather than someone's belief in the relationship, that does the causing.

Pre-emption is an important case of recursive causality, where the pre-empting causal relationship prevents the pre-empted relationship: [poisoning causing death] prevents [heart failure causing death]. Context-specific causality may also be thought of recursively: a causal relationship that only occurs in a particular context (such as susceptibility to disease amongst immune-deficient people) can often be thought of in terms of the context causing the causal relationship.

Arguably prevention is often best interpreted in terms of recursive causality. When taking mineral supplements prevents goitre, what is really happening is that taking mineral supplements prevents [poor diet causing goitre]. This is because there are other causes of goitre, such as various defects of the thyroid gland. Taking mineral supplements does not inhibit these causal chains and therefore does not prevent goitre simpliciter. In many such cases, however, the recursive nature can be eliminated by identifying a particular component of the causal chain which is prevented. Since poor diet (D) causes goitre (G) via iodine deficiency (I) and mineral supplements (S) prevent iodine deficiency, this example might be adequately represented by Figure 10.50, which is not recursive. Of course the

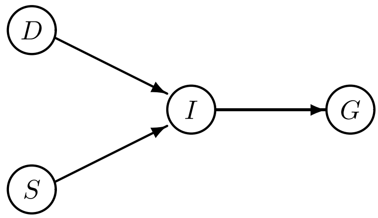


Figure 10.50: Non-recursive chain

recursive aspect can not be eliminated if no suitable intermediate variable I is known to the modeler.

We now turn our attention to Bayesian networks.

A Bayesian network is defined over a finite domain $V = \{V_1, \ldots, V_n\}$ of variables. In principle, there are no size restrictions on the set of possible values that each variable may take, but often in practice each variable will have only a finite number of possible values. For simplicity, we shall restrict our attention to two-valued variables, and denote the assignment of V_i to its values by v_i and $\neg v_i$ respectively, for $i = 1, \ldots, n$.

Herein, we say that an assignment u to a subset $U \subseteq V$ of variables is a conjunction of assignments to each of the variables in U. For example, $v_1 \wedge \neg v_2 \wedge \neg v_5$ is an assignment to $\{V_1, V_2, V_5\}$.

Definition 10.5.1 A *(causally interpreted) Bayesian network* b on V consists of two components:

- a directed acyclic graph G with nodes from V, representing the causal relations amongst the variables;
- a probability specification S, specifying, for each $V_i \in V$, the probability distribution of V_i conditional on its parents (direct causes in G), where S consists of statements of the form

  $$(p(v_i|par_i) = x_{i,par_i})$$

  for each $i = 1, \ldots, n$, such that par_i is an assignment of values to the parents of V_i and $x_{i,par_i} \in [0, 1]$. $\triangledown$

Given a Bayesian network with probability specification S, if the value of a variable V_i is known then V_i is said to be *instantiated* to that value and the corresponding probability specifiers $p(v_i|par_i)$ are 1 or 0 according to whether v_i or $\neg v_i$ is the instantiated value.

The graph and probability specification of a Bayesian network are linked by a fundamental assumption known as the *causal Markov condition*. This says that conditional on its parents, any node is probabilistically independent of all other nodes apart from its descendants, written

$$V_i \perp\!\!\!\perp ND_i \mid Par_i$$

where ND_i and Par_i are respectively the sets of non-descendants and parents of V_i.

A Bayesian network suffices to determine a joint probability distribution over its nodes, since, for each assignment v on V,

$$p(v) = \prod_{i=1}^{n} p(v_i|par_i)$$

where v_i is the assignment v gives to V_i, and par_i is the assignment v gives to the parents Par_i of V_i.

Bayesian networks are used because they offer the opportunity of an efficient representation of a joint probability distribution over V. While 2^n different probabilities $p(v)$ specify the joint distribution, these values may (depending on the structure of the causal graph G) be determined from relatively few values in the probability specification S. Furthermore, a number of algorithms have been developed for determining marginal probabilities from a Bayesian network, often very quickly. But this again depends on the structure of G (see [215] for a detailed discussion of the properties of Bayesian networks and key inference algorithms).

Causal graphs are often sparse, and thus lead to efficient Bayesian network representations. Moreover, the causal interpretation of the graph ensures that the causal Markov condition is a good default assumption, even if the conditional independence relationships it posits do not always hold in practice (see [273] on this point).

Extension to recursive causality

As noted before, causal relationships often act as causes or effects themselves. In a Bayesian network, however, the nodes tend to be thought of as simple variables, not complex causal relationships. The concept of Bayesian network can be generalized so that nodes in its causal graph G can signify complex causal relationships, while retaining the essential features of ordinary networks, namely the ability to represent joint distributions efficiently, and the ability to perform probabilistic inference efficiently.

To this end, variables can be allowed to take also Bayesian networks as values. A variable that takes Bayesian networks as values is called a *network variable* to distinguish it from a *simple variable* whose values do not contain such structure. Thus, S, which signifies "payment of subsidy to farmer" and takes value true s or false $\neg s$, is a simple variable.

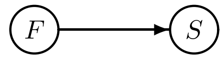


Figure 10.51: Graph of a: farming causes subsidy

But an example of a network variable is A, which stands for "agricultural policy" and takes:

- value a signifying the Bayesian network containing the graph of Figure 10.51 and the specification

$$\{p_a(f) = 0.1, p_a(s|f) = 0.9, p_a(s|\neg f) = 0.2\}$$

where F is a simple variable signifying "farming";

or

- value $\neg a$ signifying Bayesian net with graph of Figure 10.52 and specification

$$\{p_{\neg a}(f) = 0.1, p_{\neg a}(s) = 0.2\}.$$

Here, a is a policy in which farming causes subsidy and $\neg a$ is a policy in which there is no such causal relationship. For simplicity, we shall consider network variables with at most two values, but all the definitions and results referred to in the sequel also apply to network variables which take any finite number of values.

Figure 10.52: Graph of $\neg a$

Definition 10.5.2 A *recursive Bayesian network* is a Bayesian network containing at least one network variable. ▽

For example, the network with graph in Figure 10.53 and specification

$$\{p(l) = 0.7, p(a|l) = 0.95, p(a|\neg l) = 0.4\}$$

representing the causal relationship between lobbying and agricultural policy, is a recursive Bayesian network, where the simple variable L stands for "lobbying" and takes value true or false, and A is the network variable signifying "agricultural policy" discussed above.

Network variables are allowed to take recursive Bayesian networks as values. In this way a recursive Bayesian network represents a hierarchical structure.

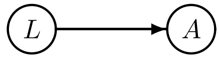


Figure 10.53: Lobbying causes agricultural policy

If a variable C is a network variable then the variables that occur as nodes in the Bayesian networks that are the values of C are called the *direct inferiors* of C, and each such variable has C as a *direct superior*. *Inferior* and *superior* are the transitive closures of these relations: thus, E is inferior to C if and only if it is directly inferior to C or directly inferior to a variable D that is inferior to C. The variables that occur in the same local network as C are called its *peers*.

A recursive Bayesian network

$$b = (G, S)$$

conveys information on a number of levels in the following way:

- the variables that are nodes in G are *level 1*;
- any variables directly inferior to level 1 variables are *level 2*, and so on;
- the network b itself can be associated with a network variable B that is instantiated to value b, and we can speak of B as the *level 0 variable*;
- the *depth* of the network is the maximum level attained by a variable.

Definition 10.5.3 A Bayesian network is said to be

- *non-recursive* if its depth is 1;
- *well-founded* if its depth is finite;
- *finite* if it is well-founded and its levels are all of finite size. ∇

We have not specified the other possible values of B: for concreteness we can suppose that B is a single-valued network variable which only takes value b. We shall restrict our discussion to finite networks.

For $i \geq 0$ let V_i be the set of level i variables, and let $\mathcal{G}_i$ and $\mathcal{S}_i$ be the set of graphs and specifications respectively that occur in networks that are values of level i variables. Thus,

- $V_0 = \{B\}$;
- $\mathcal{G}_0 = \{G\}$;
- $\mathcal{S}_0 = \{S\}$.

The domain of b is the set $V = \bigcup_i V_i$ of variables at all levels. Note that V contains the level 0 variable B itself and thus contains all the structure of b.

In our example,

$$V = \{B, L, A, F, S\}$$

where the level 0 network variable B takes value b whose graph is in Figure 10.53 and whose probability specification is

$$\{p(l) = 0.7, p(a|l) = 0.95, p(a|\neg l) = 0.4\}.$$

The only other network variable is A whose value a has the graph in Figure 10.51 and specification

$$\{p_a(f) = 0.1, p_a(s|f) = 0.9, p_a(s|\neg f) = 0.2\}$$

and whose value $\neg a$ has the graph in Figure 10.52 and specification

$$\{p_{\neg a}(f) = 0.1, p_{\neg a}(s) = 0.2\}.$$

Then V itself determines all the structure of the recursive Bayesian network in question.

A network variable V_i can be thought of as a simple variable V_i' if one drops the Bayesian network interpretation of each of its values: V_i' is the *simplification* of V_i. A recursive network b can then be interpreted as a non-recursive network b' on domain

$$V_1' = \{V_i' : V_i \in V_1\}.$$

Then, b' is called the *simplification* of b.

A variable may well occur more than once in a recursive Bayesian network, in which case it might have more than one level. Note that in a well-founded network no variable can be its own superior or inferior. A recursive Bayesian network makes causal and probabilistic claims at all its various levels, and if variables occur more than once in the network, these claims might contradict each other. We now discuss this possibility.

Network variables that occur in the domain of a recursive Bayesian network

$$b = (G, S)$$

can be interpreted as making causal and probabilistic claims about the world. Any network variable that is instantiated to a particular value asserts the validity of the network to which it is instantiated. In particular, the level 0 network variable B asserts its instantiated value b, that is, it asserts the causal relations in G, the probabilistic independence relationships one can derive from G via the causal Markov condition, and the probabilistic claims made by the probability specification S. A network variable that is not instantiated asserts the weaker claim that precisely one of its possible values is correct. A recursive Bayesian network is consistent if these claims do not contradict each other.

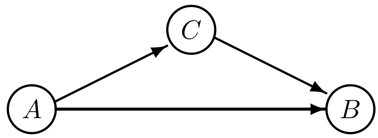


Figure 10.54: Chain $A \rightsquigarrow B$

In order to give a more precise formulation of the consistency requirement, the notion of consistency of non-recursive Bayesian networks has to be defined. There are three desiderata:

- consistency with respect to causal claims (*causal consistency*);
- consistency with respect to implied probabilistic independencies (*Markov consistency*);
- consistency with respect to probabilistic specifiers (*probabilistic consistency*).

A *chain* $A \rightsquigarrow B$ from node A to node B in a directed acyclic graph is a sequence of nodes in the graph, beginning with A and ending with B, such that there is an arrow from each node to its successor. A *subchain* of a chain c from A to B is a chain from A to B involving nodes in c in the same order, though not necessarily all the nodes in c. Thus, the chain in Figure 10.54 contains both the chain (A, C, B) and its subchain (A, B). The *interior* of a chain $A \rightsquigarrow B$ is defined as the subchain involving all nodes between A and B in the chain, not including A and B themselves.

The restriction $G_{\downharpoonleft W}$ of causal graph G defined on variables V to the set of variables $W \subseteq V$ is defined as follows: for variables $A, B \in W$, there is an arrow $A \longrightarrow B$ in $G_{\downharpoonleft W}$ if and only if $A \longrightarrow B$ is in G or, $A \rightsquigarrow B$ is in G and the variables in the interior of this chain are in $V \backslash W$. Thus, G and $G_{\downharpoonleft W}$ agree as to the causal relationships amongst variables in W. It is not hard to see that $G_{\downharpoonleft W \downharpoonleft X} = G_{\downharpoonleft X}$, for $X \subseteq W \subseteq V$.

Two causal graphs G on V and H on W are *causally consistent* if there is a third (directed and acyclic) causal graph F on $U = V \cup W$ such that $F_{\downharpoonleft V} = G$ and $F_{\downharpoonleft W} = H$. Thus, G and H are causally consistent if there is a model F of the causal relationships in both G and H. Such an F is called a *causal supergraph* of G and H.

Figure 10.55: Graph G'

The graphs G' and G'', depicted respectively in Figure 10.55 and Figure 10.56, are causally consistent because the latter graph is the restriction of the former to

$\{A, B, C\}$. However, the graph depicted in Figure 10.54 is not causally consistent with G': they do not agree as to the causal chains between A, B and C. Similarly, the graph depicted in Figure 10.54 and the graph G'' are not causally consistent.

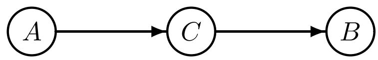


Figure 10.56: Graph G''

Note that if G and H are causally consistent and nodes A and B occur in both G and H, then there is a chain $A \rightsquigarrow B$ in G if and only if there is a chain $A \rightsquigarrow B$ in H.

Another important consistency requirement is Markov consistency. Two causal graphs G and H are *Markov consistent* if they posit (via the causal Markov condition) the same set of conditional independence relationships on the nodes they share. The graphs G' and G'' are Markov consistent because on their shared nodes A, C, B they each imply just that A and B are probabilistically independent conditional on C. The graph depicted in Figure 10.54 is not Markov consistent with either of these graphs because it does not imply this independency.

Definition 10.5.4 Two non-recursive Bayesian networks are *Markov consistent* if their causal graphs are Markov consistent. $\triangledown$

Note that Markov consistency does not imply causal consistency: for instance, two different complete graphs on the same set of nodes (a complete graph is a graph in which each pair of nodes is connected by some arrow) are Markov consistent, since neither graph implies any independence relationships, but they are not causally consistent because where they differ, they differ as to the causal claims they make. Neither does causal consistency of a pair of causal graphs imply Markov

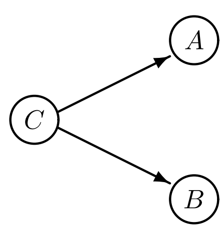


Figure 10.57: Graph H'

consistency: the graphs H' and H'', depicted respectively in Figure 10.57 and Figure 10.58, are causally consistent but H'' implies that A and B are probabilistically independent, while H' does not.

In fact, we have the following. Given a causal graph G, let $Com_G(X)$ be the set of closest common causes of X according to G, that is, the set of causes C of X

Figure 10.58: Graph H''

that are causes of at least two nodes A and B in X for which some pair of chains from C to A and C to B only have node C in common. Given causal graphs G and H on V and W respectively we say that their shared nodes are closed under closest common causes (cccc, for short) if

$$Com_G(V \cap W) \cup Com_H(V \cap W) \subseteq V \cap W.$$

Then we have the following result.

Proposition 10.5.5 *Suppose G and H are causal graphs on V and W respectively. The graphs G and H are Markov consistent if they are causally consistent and their shared nodes are closed under closest common causes.*

Proof. Suppose $X \perp\!\!\!\perp_G Y \mid Z$ for some $X, Y, Z \subseteq V \cap W$.

Recall that d-separation is a necessary and sufficient condition for deciding the conditional independences implied by a causal graph under the causal Markov condition (see [217]). For each $A \in X$ and $B \in Y$, Z *d-separates* A from B in G if (i) every chain between A and B contains a member of Z; (ii) every closest common cause of A and B is in Z, and (iii) no common effect of A and B is in Z.

Graphs G and H are causally consistent so there is a causal supergraph F on $V \cup W$. By definition, $G = F_{\downharpoonright V}$ and $H = F_{\downharpoonright W}$.

We now prove that the three d-separation conditions hold with respect to F. With respect to (i), observe that chains between A and B in G are subchains of corresponding chains in F. In what concerns (ii), A and B have the same closest common causes in G and H and, hence, in F, since F contains no nodes that are not in G and H. Finally, (iii) holds because if a common effect were in Z in F then it would also be in Z in G. Thus, $X \perp\!\!\!\perp_F Y \mid Z$. But now taking the restriction $F_{\downharpoonright W} = H$, we see that the three d-separation conditions also hold in H, for the same reasons as with the move from G to F. Thus, $X \perp\!\!\!\perp_H Y \mid Z$, as required. $\lhd$

Observe that, under the assumption of causal consistency, while closure under closest common causes is a sufficient condition for Markov consistency, it is not a necessary condition: the graph in H' in Figure 10.57 and the graph H''' in Figure 10.59 are Markov consistent because neither imply any independences just

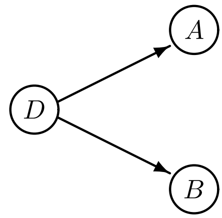


Figure 10.59: Graph H'''

amongst their shared nodes A and B, but the set of shared nodes is not closed under closest common causes.

Markov consistency is quite a strong condition. It is not sufficient merely to require that the pair of causal graphs imply sets of conditional independence relations that are consistent with each other — in fact, any two graphs satisfy this property. The motivation behind Markov consistency is based on the fact that a cause and its effect are usually probabilistically dependent conditional on the effect's other causes (this property is known as the *causal dependence condition*), in which case probabilistic independences that are not implied by the causal Markov condition are unlikely to occur.

For example, while the fact that C causes A and B (see Figure 10.57) is consistent with A and B being unconditionally independent (see Figure 10.58), it makes their independence extremely unlikely: if A and B have a common cause then the occurrence of assignment a of A may be attributable to the common cause which then renders b more likely (less likely, if the common cause is a preventative), in which case A and B are unconditionally dependent. Thus, the graphs in Figure 10.57 and Figure 10.58 are not compatible, and we need the stronger condition that independence constraints implied by each graph should agree on the set of nodes that occur in both graphs.

Finally, we turn our attention to probabilistic consistency.

Definition 10.5.6 Two causally consistent non-recursive Bayesian networks

$$(G, S) \quad \text{and} \quad (H, T)$$

defined over V and W respectively, are *probabilistically consistent* if there is some non-recursive Bayesian network (F, R), defined over $V \cup W$ and where F is a causal supergraph of G and H, whose induced probability function satisfies all the equalities in $S \cup T$. Such a network is called a *causal supernet* of (G, S) and (H, T). ▽

We now state a result concerning non-recursive Bayesian networks that are causally consistent, probabilistically consistent and closed under closest common causes.

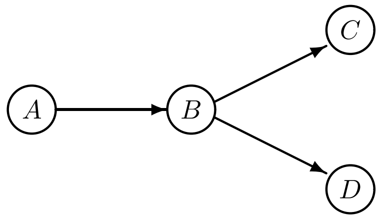


Figure 10.60: B is the closest common cause of C and D

Proposition 10.5.7 *Let (G, S) and (H, T) be two non-recursive Bayesian networks that are causally consistent, probabilistically consistent and cccc. Then there is a causal supernet (F, R) of (G, S) and (H, T) that is cccc with (G, S) and (H, T).*

Proof. Since (G, S) and (H, T) are causally and probabilistically consistent, there is a supernet (E, Q), of (G, S) and (H, T).

If E is cccc with G and H then we set $(F, R) = (E, Q)$ and we are done.

Otherwise, if E is not cccc with G, for instance, then there is some Y-structure of the form of Figure 10.60 in E, where Figure 10.61 is the corresponding struc-

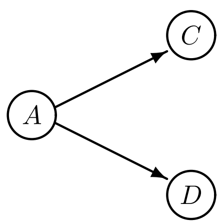


Figure 10.61: Constructing F

ture in G, where in these diagrams we take the arrows to signify the existence of causal chains rather than direct causal relations. Note that B must be in G or H, since the domain of a causal supergraph of G and H is the union of the domains of G and H, and B cannot be in G since otherwise, by causal consistency, the chain from A to C in G would go via B. Hence, B is in H. Note also that not both of C and D can be in H, for otherwise G and H are not cccc. Suppose then that D is not in H. Then, the chain from B to D is not in G or H. Construct F by taking E, removing the chain from B to D and including a chain from A to D, as in Figure 10.62. Do this for all such Y-structures not replicated in G. F remains a causal supergraph of G and H, since the chain from B to H was redundant. Moreover, F is now cccc with G.

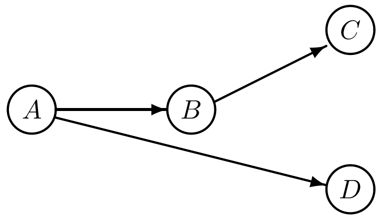


Figure 10.62: A is the closest common cause of C and D

Next, construct the associated probability specification R by determining specifiers from (E, Q). Thus, if the causal chain from A to D is direct we can set

$$p(d|a) = \sum_b p_{(E,Q)}(d|b) p_{(E,Q)}(b|a)$$

in R. It is not hard to see that $p_{(F,R)} = p_{(E,Q)}$ so the new network is also a causal supernet of (G, S). If E is not cccc with H then repeat this algorithm, to yield a causal supernet of (G, S) and (H, T) that is cccc with (G, S) and (H, T). $\triangleleft$

Note that the requirement that G and H are cccc in the above result is essential. If G is as in Figure 10.60 and H is as in Figure 10.61, then there is no causal supergraph of G and H that is cccc with G and H.

Proposition 10.5.8 *Consider two non-recursive Bayesian networks causally consistent, probabilistically consistent and cccc. Then they determine the same probability function over the variables they share.*

Proof. Suppose (G, S) and (H, T) are causally and probabilistically consistent and cccc. Then, by Proposition 10.5.7, there is a causal supernet (F, R) that is cccc with both nets. By Proposition 10.5.5, F is Markov consistent with G and H.

Next, note that (G, S) and (F, R) determine the same probability function over variables V of (G, S), that is $p_{(G,S)}(v) = p_{(F,R)}(v)$. Indeed, using the fact that (F, R) is a causal supernet of (G, S),

$$\begin{aligned} p_{(G,S)}(v) &= \prod_{v_i \in V} p_{(G,S)}(v_i | par_i^G) \\ &= \prod_{v_i \in V} p_{(F,R)}(v_i | par_i^G) \\ &= \prod_{v_i \in V} p_{(F,R)}(v_i | v_1, \ldots, v_{i-1}) \\ &= p_{(F,R)}(v) \end{aligned}$$

where par_i^G is the state of the parents of V_i according to G that is consistent with assignment v to V, and where it is supposed that the variables $V_1, \ldots, V_n$ in V are ordered G-ancestrally, that is, no descendants of V_i in G occur before V_i in the order. Observe also that $V_i \perp\!\!\!\perp_G V_1, \ldots, V_{i-1} \mid Par_i^G$ implies $V_i \perp\!\!\!\perp_F V_1, \ldots, V_{i-1} \mid Par_i^G$ by Markov consistency.

Similarly, (H, T) and (F, R) determine the same probability function over the variables of (H, T). Hence, (G, S) and (H, T) determine the same probability function over variables they share. $\triangleleft$

Because Proposition 10.5.8 is a desirable property in itself, closure under closest common causes is adopted as a consistency condition.

Definition 10.5.9 Two non-recursive networks are *consistent* if they are causally and probabilistically consistent and cccc. $\triangledown$

By Proposition 10.5.5 consistency implies Markov consistency. Hence, we can state what it means for a recursive network to be consistent.

An assignment v of values to variables in V, the domain of a recursive Bayesian network b, assigns values to all the simple variables and network variables that occur in b. Take, for instance, the recursive Bayesian network b of Figure 10.53: therein,

$$V = \{B, L, A, F, S\}$$

and

$$b \wedge l \wedge (\neg a) \wedge f \wedge (\neg s)$$

is an example of an assignment to V. Note that the level 0 variable B only takes one value b and so must always be assigned this value. Consider the assignment of values given to network variables in V. In our example, the network variables are B and A and these are assigned values b and $\neg a$ respectively. Each such value is itself a recursive Bayesian network, and when simplified induces a non-recursive Bayesian network. Let $\underline{b}_v$ denote the set of recursive Bayesian networks induced by v and let $\underline{b}'_v$ denote the set of non-recursive Bayesian networks formed by simplifying the networks in $\underline{b}_v$.

Assignment v is *consistent* if each pair of networks in $\underline{b}'_v$ is consistent (that is, if each pair of values of network variables is consistent, when these values are interpreted non-recursively).

Definition 10.5.10 A recursive Bayesian network is *consistent* if it has some consistent assignment v of values to V. $\triangledown$

Thus, if a recursive Bayesian network is not to be self-contradictory there must be some assignment under which all pairs of network variables satisfy three regularity conditions: causal consistency, probabilistic consistency and closure under closest common causes.

Note that it is easy to turn a recursive network into one that is causally consistent, by ensuring that causal chains correspond for some assignment, and then cccc (and so Markov consistent), by ensuring that shared nodes of pairs of graphs also share closest common causes, for some assignment.

In order to make G_2 in Figure 10.64 causally consistent with graph G_1 of Figure 10.63, for example, we need to introduce a chain that corresponds to the chain

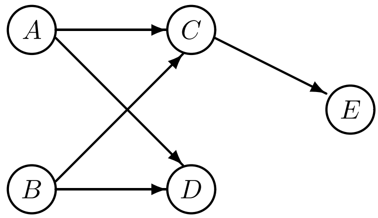


Figure 10.63: Graph G_1

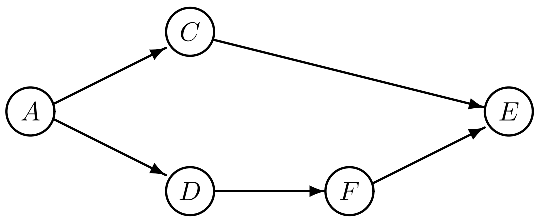


Figure 10.64: Graph G_2

(D, F, E) in G_2, by adding an arrow from D to E in G_1. In order to make G_2 and G_1 cccc (and so Markov consistent) we need to add B to G_2 as a closest common cause of C and D. The modified graphs are depicted in Figure 10.65 and Figure 10.66.

Similarly, in practice, one would not expect each probability specification to be provided independently and then to have the problem of checking consistency — one would expect to use conditional distributions in one specification to determine distributions in others. For example, a probability specification on H_2

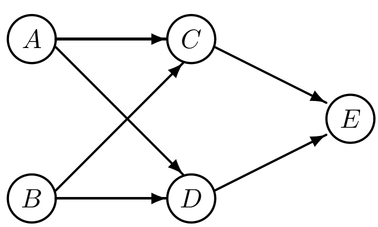


Figure 10.65: Graph H_1

in Figure 10.66 would completely determine a probability specification on H_1 in Figure 10.65.

We turn our attention to joint distributions. Any non-recursive Bayesian network is subject to the causal Markov condition which determines a joint probability

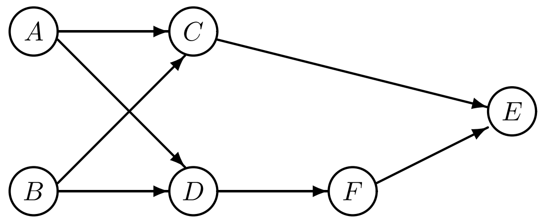


Figure 10.66: Graph H_2

distribution over the variables of the network from its graph and probability specification. We shall suppose that recursive Bayesian networks also satisfy the causal Markov condition.

A recursive Bayesian network contains network variables whose values are interpreted as (recursive or non-recursive) Bayesian networks. Thus, a recursive Bayesian network suffices to determine a hierarchy of joint probability distributions p_a on the (level 1) variables for each a that occurs as the value of a network variable.

Standard Bayesian network algorithms can be used to perform inference in a recursive Bayesian network, and the range of causal-probabilistic questions that can be addressed is substantially increased. For example, one can answer questions like "what is the probability of a subsidy given farming?" (see Figure 10.51) and "what is the probability of lobbying given agricultural policy ($\neg a$)?" (see Figure 10.53).

But certain questions remain unanswered. We can not as yet determine the probability of one node conditional on another if the nodes only occur at different levels of the network. For example we can not answer the question "what is the probability of subsidy given lobbying?" While we have a hierarchy of joint distributions, we have not yet specified a single joint distribution over the set of nodes in the union of the graph, that is, over the recursive network as a whole.

In fact, a recursive network does determine such an over-arching joint distribution considering an extra independence assumption, called the *recursive Markov condition*: each variable is probabilistically independent of those other variables that are neither its inferiors nor its peers, conditional on its direct superiors.

A precise explanation of the causal Markov condition and recursive Markov condition is given in the sequel.

Given a recursive Bayesian network domain V and a consistent assignment v of values to V, we construct a non-recursive Bayesian network, $v^{\downarrow}$, called *flattening* of v, as follows:

- the domain of $v^{\downarrow}$ is V;
- the graph $G^{\downarrow}$ of $v^{\downarrow}$ has variables in V as nodes, each variable occurring only once in the graph;
- add an arrow from V_i to V_j in $G^{\downarrow}$ if

– V_i is a parent of V_j in v (that is, there is an arrow from V_i to V_j in the graph of some value of v)

or

– V_i is a direct superior of V_j in v (that is, V_j occurs in the graph of the value that v assigns to V_i).

We will describe the probability specification $S^{\downarrow}$ of $v^{\downarrow}$ in due course. First, we refer to some properties of the graph $G^{\downarrow}$.

Note that $G^{\downarrow}$ may or may not be acyclic. If we take our farming example

$$V = \{B, L, A, F, S\}$$

presented above then the graph of the flattening

$$(b \wedge \neg l \wedge a \wedge f \wedge s)^{\downarrow}$$

is depicted in Figure 10.67 and is acyclic.

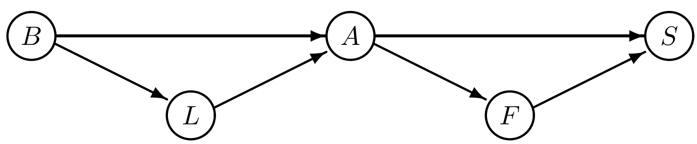


Figure 10.67: Example of flattening

But the graph of the flattening of assignment $b \wedge c \wedge d \wedge e$ to $\{B, C, D, E\}$, where B is the level 0 network variable whose value b has graph $C \longrightarrow D$, C and E are simple variables and D is a network variable whose assigned value d has the graph $E \longrightarrow C$, is cyclic.

The graph in a non-recursive Bayesian network must be acyclic in order to apply standard Bayesian network algorithms, and this requirement extends to recursive Bayesian networks: we will focus on consistent *acyclic* assignments to a recursive Bayesian network domain, those consistent assignments v that lead to an acyclic graph in the flattening $v^{\downarrow}$.

By focusing on consistent acyclic assignments v, the following explanations of the two independence conditions become plausible. Given a consistent acyclic assignment v, let:

- PND_i^v be the set of variables that are peers but not descendants of V_i in v;
- NIP_i^v be the non-inferiors or peers of V_i;
- $DSup_i^v$ be the direct superiors of V_i.

As before, Par_i^v are the parents of V_i and ND_i^v are the non-descendants of V_i. None of these sets are taken to include V_i itself.

Causal Markov Condition (CMC)
For each $i = 1, \ldots, n$ and $DSup_i^v \subseteq X \subseteq NIP_i^v$,

$$V_i \perp\!\!\!\perp PND_i^v \mid Par_i^v, X.$$

Recursive Markov Condition (RMC)
For each $i = 1, \ldots, n$ and $Par_i^v \subseteq X \subseteq PND_i^v$,

$$V_i \perp\!\!\!\perp NIP_i^v \mid DSup_i^v, X.$$

The graph of the flattening has the following property.

Proposition 10.5.11 *Let v be a consistent acyclic assignment to a recursive Bayesian network domain V. Then the probabilistic independences implied by v via the causal Markov condition and the recursive Markov condition are just those implied by the graph $G^{\downarrow}$ of the flattening $v^{\downarrow}$ via the causal Markov condition.*

Proof. Order the variables in V ancestrally with respect to $G^{\downarrow}$, that is, no descendants of V_i in $G^{\downarrow}$ occur before V_i in the ordering. Note that this is always possible because $G^{\downarrow}$ is acyclic.

First we shall show that CMC and RMC for v imply CMC for $G^{\downarrow}$. By Corollary 3 of [217] it suffices to show that $V_i \perp\!\!\!\perp V_1, \ldots, V_{i-1} \mid Par_i^{G^{\downarrow}}$ for any $V_i \in V$. By CMC,

$$V_i \perp\!\!\!\perp PND_i^v \mid Par_i^v, DSup_i^v$$

and by RMC,

$$V_i \perp\!\!\!\perp NIP_i^v \mid DSup_i^v, PND_i^v.$$

The following property of probabilistic independence holds ([217]): if $R \perp\!\!\!\perp S|T$ and $R \perp\!\!\!\perp U|S, T$ then $R \perp\!\!\!\perp S, U|T$. Using this property,

$$V_i \perp\!\!\!\perp PND_i^v \cup NIP_i^v \mid Par_i^v, DSup_i^v.$$

Now $\{V_1, \ldots, V_n\} \subseteq PND_i^v \cup NIP_i^v$ since the variables are ordered ancestrally and v is acyclic, and the parents of V_i in $G^{\downarrow}$ are just the parents and direct superiors of V_i in v,

$$Par_i^{G^{\downarrow}} = Par_i^v \cup DSup_i^v$$

so

$$V_i \perp\!\!\!\perp V_1, \ldots, V_{i-1} \mid Par_i^{G^{\downarrow}}$$

as required.

Next we shall see that CMC for $G^{\downarrow}$ implies CMC and RMC for v. In fact, this follows straightforwardly by d-separation (see the proof of Proposition 10.5.5). $Par_i^v \cup X$ d-separates V_i and PND_i^v in $G^{\downarrow}$ for any $DSup_i^v \subseteq X \subseteq NIP_i^v$, since $Par_i^v \cup X$ includes the parents of V_i in $G^{\downarrow}$ and, by acyclicity of v, PND_i^v are non-descendants of V_i in $G^{\downarrow}$, so CMC holds.

$$DSup_i^v \cup X \text{ } d\text{-separates } V_i \text{ and } NIP_i^v \text{ in } G^{\downarrow}$$

for any $Par_i^v \subseteq X \subseteq PND_i^v$, since $DSup_i^v \cup X$ includes the parents of V_i in $G^{\downarrow}$ and, by acyclicity of v, NIP_i^v are non-descendants of V_i in $G^{\downarrow}$, so RMC holds. ◁

We now define the probability specification $S^{\downarrow}$ of the flattening $v^{\downarrow}$. In the specification $S^{\downarrow}$ we need to provide a value for $p(v_i|par_i^{G^{\downarrow}})$ for each value v_i of V_i and assignment $par_i^{G^{\downarrow}}$ of the parents $Par_i^{G^{\downarrow}}$ of V_i in $G^{\downarrow}$. If V_i only occurs once in the recursive Bayesian network determined by v then we can define

$$p(v_i|par_i^{G^{\downarrow}}) = p(v_i|dsup_i^v \wedge par_i^v) = p_{dsup_i^v}(v_i|par_i^v),$$

which is provided in the specification of the value of V_i's direct superior in v. If V_i occurs more than once in the recursive Bayesian network determined by v then the specifications of v contain $p_{dsup_i^G}(v_i|par_i^G)$ for each graph G in v in which V_i occurs. Then

$$DSup_i^v = \bigcup_G DSup_i^G$$

and

$$Par_i^v = \bigcup_G Par_i^G$$

with the unions taken over all such G. Now the specifiers $p_{dsup_i^G}(v_i|par_i^G)$ constrain the value of $p_{dsup_i^v}(v_i|par_i^v)$ but may not determine it completely. These are linear constraints, though, and thus there is a unique value for $p_{dsup_i^v}(v_i|par_i^v)$ which maximizes entropy subject to the constraints holding. This can be taken as its optimal value (see [157]) and $p(v_i|par_i^{G^{\downarrow}})$ can be set to this value (see [274] for more on maximizing entropy).

Having fully defined the flattening $v^{\downarrow} = (G^{\downarrow}, S^{\downarrow})$ and shown that the causal Markov condition holds, we have a (non-recursive) Bayesian network, which can be used to determine a probability function over assignments to v. As usual, v_i is the value v assigns to V_i and $par_i^{G^{\downarrow}}$ is the assignment v gives to the parents of V_i according to $G^{\downarrow}$.

Proposition 10.5.12 *A recursive Bayesian network determines a unique joint distribution over consistent acyclic assignments v of values to its domain, defined by*

$$p(v) = \prod_{i=1}^{n} p(v_i|par_i^{G^{\downarrow}})$$

where $G^{\downarrow}$ is the graph in the flattening $v^{\downarrow}$ of v and $p(v_i|par_i^{G^{\downarrow}})$ is the value in the specification $S^{\downarrow}$ of $v^{\downarrow}$.

Observe that the domain of p is the set of assignments to V, and p is unique over consistent acyclic assignments. If one wants to take just the set of consistent acyclic assignments as domain of p (equivalently, to award probability 0 to inconsistent or cyclic assignments) then one must renormalize, that is, divide $p(v)$ by $\sum p(v)$ where the sum is taken over all consistent acyclic assignments.

While a flattening is a useful concept to explain how a joint distribution is defined, there is no need to actually construct flattenings when performing calculations with recursive networks. Indeed that would be most undesirable, given that there are exponentially many assignments and thus exponentially many flattenings which would need to be constructed and stored. By Proposition 10.5.12, only the probabilities

$$p(v_i|par_i^v \wedge dsup_i^v)$$

need to be determined, and in many cases (when V_i occurs only once in v) these are already stored in the recursive network.

The concept of flattening, in which a mapping is created between a recursive network and a corresponding non-recursive network, also helps us understand how standard inference algorithms for non-recursive Bayesian networks can be directly applied to recursive networks.

For example, message-passing propagation algorithms (see [217] [215]) can be directly applied to recursive networks, as long as messages are passed between direct superior and direct inferior as well as between parent and child. Moreover, recursive Bayesian networks can be used to reason about interventions just as can non-recursive networks: when one intervenes to fix the value of a variable one must treat that variable as a root node in the network, ignoring any connections between the node and its parents or direct superiors (see [218]). In effect, tools for handling non-recursive Bayesian networks can be easily mapped to recursive networks.

A word on the plausibility of the recursive Markov condition. It was shown in [273] that the causal Markov condition can be justified as follows: suppose an agent's background knowledge consists of the components of a causally interpreted Bayesian network – knowledge of causal relationships embodied by the causal graph and knowledge of probabilities encapsulated in the corresponding probability specification – then the agent's degrees of belief ought to satisfy the causal Markov condition (see also [274]).

This justification rests on the acceptance of the maximum entropy principle (which says that an agent's belief function should be the probability function, out of all those that satisfy the constraints imposed by background knowledge, that has maximum entropy) and the causal irrelevance principle (which says that if an agent learns of the existence of new variables which are not causes of any of the old variables, then her degrees of belief concerning the old variables should not change). An analogous justification can be provided for the recursive Markov condition.

Clearly, learning of new variables that are not superiors (or causes) of old variables should not lead to any change in degrees of belief over the old domain. Now if an agent's background knowledge takes the form of the components of a recursive Bayesian network then the maximum entropy function, and thus the agent's degrees of belief, will satisfy the recursive Markov condition as well as the causal Markov condition. Thus, a justification can be given for both the causal Markov condition and the recursive Markov condition.

10.6 Self-fibring networks

In this section we concentrate on the subject of self-fibring of networks. We show that the recursive network approach considered fits within a more general concept of substituting one network inside another (referred to as self-fibring of networks). We focus our attention on *information networks*, which are directed graphs whose roots are *inputs*, whose leaves are *outputs* and whose arrows indicate the flow of information from input to output.

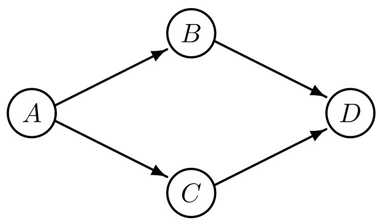


Figure 10.68: The graph of a Bayesian network

An application of a Bayesian network, for example, can be construed as an information network as follows. When a Bayesian network is applied, the values of a set of variables are observed. These variables are the inputs. They are instantiated to their observed values in the Bayesian network, and this change is propagated around the network, typically using message-passing algorithms, (see [217]), until the probabilities of further variables of interest (the outputs) can be ascertained. Thus, in message-passing algorithms, information flows from the inputs to the outputs via the arrows of the Bayesian network, though not normally in accordance with the direction of the arrows in the original Bayesian network.

Suppose, for instance, that Figure 10.68 is the graph of a Bayesian network, that the value of B is observed and that the probability of C is required. Then, in determining the probability of C, information flows from B to C along the pathways between B and C of the original Bayesian network graph, as depicted in Figure 10.69.

Note that, in general, the information network is only a schematic representation of the flow of information: in fact, in message-passing propagation algorithms, messages are passed in both directions along arrows, two passes are made of the network, and in multiply connected graphs, such as the one in Figure 10.68, propagation takes place in an associated undirected tree-shaped Markov network formed from the Bayesian network (see [177]). In singly-connected Bayesian networks,

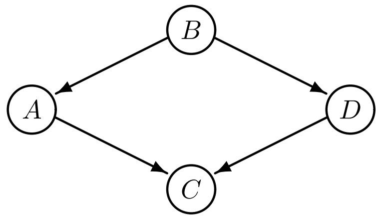


Figure 10.69: The graph of a corresponding information network

though, there is a fairly close correspondence between information network and flow of messages.

The question now arises as to how information networks can be *self-fibred*, that is, substituted one inside the other. There are several options for self-fibring. We now explain them briefly and present the full definitions later on.

Let $\boldsymbol{B}(X)$ be a network with node X in it. Let $\boldsymbol{A}$ be another network. We want to define $\boldsymbol{C} = \boldsymbol{B}(X/\boldsymbol{A})$, a new network which is the result of substituting $\boldsymbol{A}$ for X. Already at this stage there are several views to take.

- *View 1: Syntactical Substitution*
  Regard the operation at the syntactical level. Define $\boldsymbol{C}$ syntactically and give it meaning / semantics / probabilities derived from the meanings of $\boldsymbol{B}$ and $\boldsymbol{A}$.

- *View 2: Semantic Insertion*
  Look at the meaning of $\boldsymbol{B}$ and then define what $\boldsymbol{B}(X/\boldsymbol{A})$ is supposed to be. Here the substitution is not purely syntactic. For example, if $\boldsymbol{B}$ is a Bayesian network where the node X can take two values 0, 1 then if X is 0 we substitute (in a certain way) $\boldsymbol{A}_0$ for X and if X is 1 we substitute $\boldsymbol{A}_1$. The "substitution" need not be actual substitution but some operation $\mathit{Ins}(X, \boldsymbol{B}, \boldsymbol{A})$ inserting $\boldsymbol{A}_i$ at the point X inside $\boldsymbol{B}$.

So, for example, in logic we can have

$$\mathit{Ins}(X, X \to B, C) \stackrel{\text{def}}{=} (X \to C) \to B.$$

Thus, in this case,

$$\mathit{Ins}(X, \boldsymbol{B}(X), C) = \boldsymbol{B}(X/X \to C).$$

More complex insertions are possible for Bayesian nets. We could convert in the above case the semantic inversion into a syntactic one by splitting each variable Y in the net into two variables Y_0 (for $Y = 0$) and Y_1 (for $Y = 1$).

Recall that by a network we mean a finite directed acyclic graph $G = (V, R)$ with nodes from V and where R is the (immediate) parent relation.

Recall Figure 10.50. A network can be used for input output as follows:

- the variables of the network can range over a domain D of possible inputs;
- the arrows can be interpreted as indicating propagation or processing of the inputs from the parent nodes into the child.

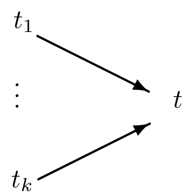


Figure 10.70: Network propagating t_1 and t_k to t

Thus, if we have the network in Figure 10.70 then we propagate the input from t_i into t. If $\boldsymbol{V}(x)$ is the value at node x, then we need a propagation function f yielding

$$\boldsymbol{V}(t) = f(\boldsymbol{V}(t_1), \ldots, \boldsymbol{V}(t_n)).$$

Note that there may be a constraint $\phi(t_1, \ldots, t_n)$ on the inputs: only if a set of value of inputs satisfies ϕ will those values be admissible.

In Bayesian networks, the arrows correspond to causal direction rather than to the flow of information. But in an information network constructed from a Bayesian network, the arrows correspond to flow of information. Thus, we focus on information networks constructed from Bayesian networks henceforth. Here, the function f of interest is the probability distribution of the outputs conditioned on the observed assignment to the inputs,

$$p(t|t_1, \ldots, t_n).$$

Our aim is to look at the arrows as implications with a view of giving meaning to substituting networks within networks (self-fibring of networks). To illustrate the ideas of self-fibring we begin with the simple two point network $\mathcal{N}_1$ depicted in Figure 10.71. The input gives value to A and this is propagated to B, using the

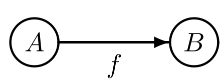


Figure 10.71: Network $\mathcal{N}_1$

function f. We now give several interpretations for this as implication.

- *Interpretation 1*
  The above represents a substructural implication $A \to B$. The semantical

interpretation for the substructural $\rightarrow$ is via evaluation into an algebraic semigroup $(S, \circ, e)$, where $\circ$ is a binary associative operation and e is the identity. If the formula $A \rightarrow B$ gets value t and the input A get a value a then B gets value $b = t \circ a$. Here the network function f can be taken as the function

$$\lambda x f_t(x) = \lambda x(t \circ x).$$

- *Interpretation 2*
  This interpretation is the modus ponens in a Labeled Deductive System. The rule has the form

$$\frac{\alpha : A,\ \beta : A \rightarrow B,\ \varphi(\beta, \alpha)}{\boldsymbol{f}(\beta, \alpha) : B}.$$

  Its meaning is that if we prove A with label α and $A \rightarrow B$ with label β and (β, α) satisfy the enabling condition φ, then we can deduce B with label

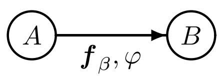


Figure 10.72: Network with labeled implication

  $\boldsymbol{f}(\beta, \alpha)$ and we have:

$$\lambda x \boldsymbol{f}_\beta(x) = \lambda x \boldsymbol{f}(\beta, x).$$

  The side condition φ is $\varphi([a, b], [c, d])$ is that $k \neq 1$. Thus, to interpret the labeled implication in our network we need to add φ to the link as in Figure 10.72.

- *Interpretation 3*
  In this case we see intuitionistic formulas as types. The formulas $A, B, A \rightarrow B$ are understood as λ calculus types having λ terms inhabiting them. We read $\mathcal{N}_1$ of Figure 10.71 as a network which, for any term t of type A given as input, the network outputs the $f(t)$ term of type B. Thus, f is of type $A \rightarrow B$.

- *Interpretation 4*
  We can regard $\mathcal{N}_1$ as a causal Bayesian network. The variable A can take states $a_1, \ldots, a_k$ and the variable B can take states $b_1, \ldots, b_m$. Then, the table f must give the conditional probability $P(B|A)$, giving the probability $p_{i,j}$ of B being in the state b_j, given that A is in the state a_i. We must have $\sum_j p_{ij} = 1$. The matrix is

$$P = \begin{pmatrix} P_{11} & P_{n1} \\ \vdots & \vdots \\ P_{k1} & P_{mk} \end{pmatrix}$$

If q_i is the probability of A being in the state a_i, ($\Sigma q_i = 1$) then the probability of B being in the state b_j is $b_j = \sum_i p_{ij} q_i$. Thus,

$$(b_1, \ldots, b_n) = (a_1, \ldots, a_k) \cdot P.$$

We can take f as the matrix $P = (p_{ij})$ and the input as $\overrightarrow{a} = (a_1, \ldots, a_k)$ and then the output is

$$\overrightarrow{b} = \overrightarrow{a} P.$$

The logical interpretations (1)–(3) allow us to give meaning to self-fibred networks where we substitute a network within a network. Here we have the options of syntactical substitution (view 1) or semantical insertion (view 2). We choose semantical insertion, where we have one insertion for $X = 1$ and another for $X = 0$. The insertion makes X a parent to all nodes in the substituted network.

The Dempster-Shafer rule is a special case of interpretation (2). The Dempster-Shafer set up allows for certainty values for $A, B, A \to B$, to be closed intervals of real numbers. Thus, if A has value in the real closed interval $[a, b]$ and the implication $A \to B$ has value in the interval $[c, d]$, then B has value in the interval

$$[a, b] \circ [c, d] = \left[\frac{ad + bc - ac}{1 - k}, \frac{bd}{1 - k} \right]$$

with $k = a(1 - d) + c(1 - b)$.

The network in Figure 10.73 for $X = 0$ and the network in Figure 10.74 for $X = 1$ show how this works for $\boldsymbol{B}(X) = X \to C$ and $\boldsymbol{A}_0 = A \to B$. We use

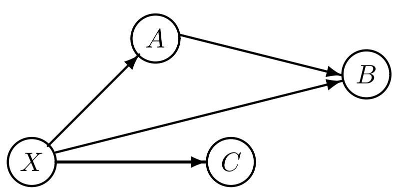


Figure 10.73: Network for $X = 0$

$X \twoheadrightarrow \boldsymbol{A}$ to denote that X connects directly to all elements in $\boldsymbol{A}$.

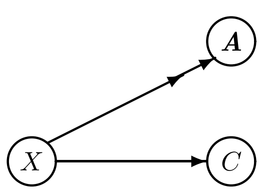


Figure 10.74: Network for $X = 1$

We can adopt a view closer to that of logic and have no insertion if $X = 0$ and yes insertion in case $X = 1$ of a single network. The simplest cases are depicted in Figure 10.75.

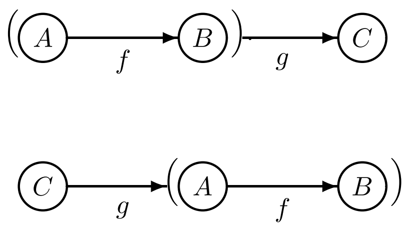


Figure 10.75: Simple cases of fibred networks

In the first case, we took the network $\mathcal{N}_2$ depicted in Figure 10.76 and substitute for X the network $\mathcal{N}_1$ (recall Figure 10.71). The second case is similar.

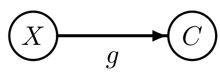


Figure 10.76: Network $\mathcal{N}_2$

The question is what meaning do we give to these fibred networks?

Let us consider the first case (see Figure 10.77). The machinery underlying the

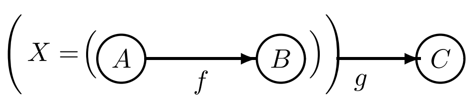


Figure 10.77: Example of substitution

network $\mathcal{N}_2$ is to accept inputs x of a certain kind at the node X and output $g(x)$ at node C. By letting $X = A \to B$ we must ask: what is the input we are getting for X? We give the obvious answer, saying that the input is f. We now have to check whether f is of the kind that can be accepted in our network.

There is no problem when we consider the interpretations (1) to (3) above.

In fact, in (1) f can be identified with an element t, and in (2) f can be identified with a label. In (3), f is a λ-term of type $A \to B$ and we can say that X (accepts elements) of type $A \to B$ and g is of type $(A \to B) \to C$.

However, in interpretation (4), A is a probability distribution for the states of A and f is a matrix of conditional probabilities. First let us simplify and say both A and B are two state variables $A = 0, A = 1, B = 0, B = 1$. Even with this simplification, still f is a 2×2 matrix P. It allows for many states not only just two. As a consequence, we have a problem in this case.

To solve this problem, we can allow for new kinds of inputs for our variables. This option is complicated because of repeated iteration of fibring and it will not be pursued in the sequel.

Another possibility is to extract from the new input (the matrix) a recognizable input for X in $X \to C$ (that is, a two state input). This method is what we usually do in the area of fibring logics.

We need a fibring function $\boldsymbol{F}$ that will extract two states, “yes” or “no”, out of the matrix P: “yes” if B depends on A in any way and “no” otherwise. In other words, we read X as a variable getting 1 if the network substituted for it is “on” or “active” and 0 if it is not “on”.

Hence, for example, if the matrix is

$$\begin{pmatrix} p & 1-p \\ p & 1-p \end{pmatrix}$$

we get

$$(q, 1-q) \begin{pmatrix} p & 1-p \\ & \\ p & 1-p \end{pmatrix} = (p, 1-p)$$

and thus the probability of B is independent of that of A.

Say we have, for $0 \le \varepsilon \le 1-p$,

$$(q, 1-q) \begin{pmatrix} p & 1-p \\ & \\ p+\varepsilon & 1-p-\varepsilon \end{pmatrix} =$$

$$(pq + (1-q)p + (1-q)\varepsilon, (1-p)q + (1-q)(1-p) - (1-q)\varepsilon) =$$

$$(p + (1-q)\varepsilon, (1-p) - (1-q)\varepsilon).$$

The variation is $2\varepsilon(1-q) \le 2\varepsilon$. So we can give a probability for $\varepsilon = 0$ or $\varepsilon \ne 0$.

We leave this aspect for a moment and discuss the other possibility of fibring, namely, the second case in Figure 10.76. Here, we substitute the network $\mathcal{N}_1$ for the variable Y in the network $\mathcal{N}_3$ depicted Figure 10.78.

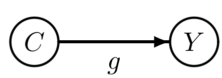


Figure 10.78: Network $\mathcal{N}_3$

The first three interpretations will cope with this very well, because the output of g can modify the f, since they are of the same kind. Can we do something similar in the probabilities case? We again have several options.

A first option consists in reading Y as a variable getting values in $\{0, 1\}$ indicating whether the network $\mathcal{N}_1$ is “on”' or not. The value of Y is obtained in the network $\mathcal{N}_3$.

A second option consists in using the network up to Y to modify the network which we substitute for Y. Observe that in case of neural networks this is the more reasonable option.

We note that g is a matrix and so is f. Should we modify f by multiplying it by g and set something like we see in Figure 10.79?

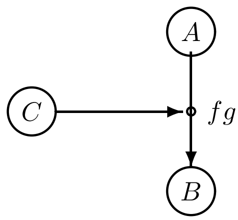


Figure 10.79: Multiplying f by g

How would this relate to the network $\mathcal{N}_4$ depicted in Figure 10.80?

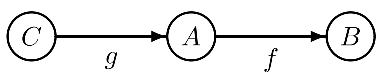


Figure 10.80: Network $\mathcal{N}_4$

Let us check that the network $\mathcal{N}_1$ has the matrix

$$\begin{pmatrix} p_1, & 1-p_1 \\ p_2, & 1-p_2 \end{pmatrix}$$

and that the network $\mathcal{N}_3$ has the matrix

$$\begin{pmatrix} \gamma_1, & 1-\gamma_1 \\ \gamma_2, & 1-\gamma_2 \end{pmatrix}$$

The product gf is

$$\begin{pmatrix} \gamma_1 & 1-\gamma_1 \\ \gamma_2 & 1-\gamma_2 \end{pmatrix} \cdot \begin{pmatrix} P_1 & 1-P_1 \\ P_2 & 1-P_2 \end{pmatrix} =$$

$$\begin{pmatrix} \gamma_1 p_1 + (1-\gamma_1)p_2, & \gamma_1(1-p_1) + (1-\gamma_1)(1-p_2) \\ \gamma_2 1 p_1 + (1-\gamma_2)p_2, & \gamma_2(1-p_1) + (1-\gamma_2)(1-p_2) \end{pmatrix}$$

This would interpret the second case of fibred network in Figure 10.75 as the network $\mathcal{N}_4$. We prefer the first option.

Let us now see what to do with networks of the form of the network $\mathcal{N}_5$ depicted in Figure 10.81.

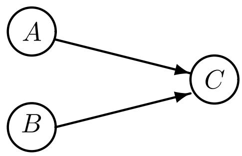


Figure 10.81: Network $\mathcal{N}_5$

Can we read the symbol $\rightarrow$ as implication? The answer is "yes" for the first three logical interpretations. We read it as

$$\langle A, B \rangle \rightarrow C$$

or

$$A \otimes B \rightarrow C$$

where $\otimes$ is a commutative binary operation. It is the multiplicative conjunction in linear logic and is the ordinary conjunction in intuitionistic logic. We have in the case of logic that:

$$(A \otimes B \rightarrow C) \equiv (A \rightarrow (B \rightarrow C)) \equiv (B \rightarrow (A \rightarrow C)).$$

For the Dempster-Shafer rule we calculate $[a, b] \otimes [c, d]$ as $[a, b] \circ [c, d]$.

This does not hold in the Bayesian network case. We need a function giving a probability value for C, for each pair of possible values (x, y) for (A, B).

We still need to give meaning to the two fibred networks depicted in Figure 10.82.

The first one is obtained by substituting the network $\mathcal{N}_5$ for X in the network $\mathcal{N}_6$ in Figure 10.83. The second one is obtained substituting the network $\mathcal{N}_5$ for X in the network $\mathcal{N}_7$ depicted Figure 10.84.

The principles we discovered still hold. In the first case the fibring function gives X values 0 or 1 depending whether we believe in the connection between A, B, C, that is, the network is "on" or not.

The second case would require modifying the network of A, B, C by using the network $\mathcal{N}_7$.

The simplest is to take the first option mentioned above. The value $X = 1$ means the network with A, B, C is "on" and otherwise it is not.

In any case, the kind of choices we have to make are clear! There is a lot of scope for fine tuning. For example, we can look at the network depicted in Figure 10.85 as a family of networks of the form depicted in Figure 10.86 using the probabilities in the substituted network (fix A_j, $j \neq i$ as 0,1) to decide on priorities.

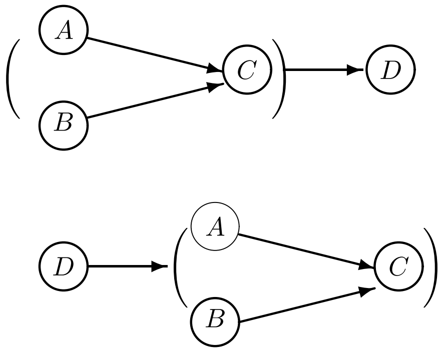


Figure 10.82: Two more cases of fibred networks

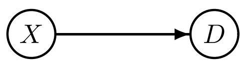


Figure 10.83: Network $\mathcal{N}_6$

We now present a general theory of networks and fibring of networks. We begin with the definition of network.

Definition 10.6.1 A *network* is a tuple

$$\mathcal{N} = (S, R, D, E, \tau, V, L, \mathbb{F}, \Omega)$$

where

- S is the set of nodes;
- $R \subseteq S \times S$;
- $D \subseteq S$, $D \neq \emptyset$, is the set of input nodes;
- $E \subseteq S$, $E \neq \emptyset$, is the set of output nodes;
- $\tau : R \mapsto L$ is the labeling function;
- L is the set of labels;
- $V : S \mapsto L$ is the coloring function;
- $\mathbb{F}$ is a function giving some value in the space Ω to any finite list of the form $(t, (V_1, l_1), \ldots, (V_n, l_n))$ in such way that

$$V(t) = \mathbb{F}(t, (V(x_1), \tau(x_1, t)), \ldots, (V(x_n), \tau(x_n, t)))$$

for any $t \in S$ and x_i such that $x_i R t$, where x_i are all the parents of t;

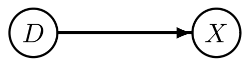


Figure 10.84: Network $\mathcal{N}_7$

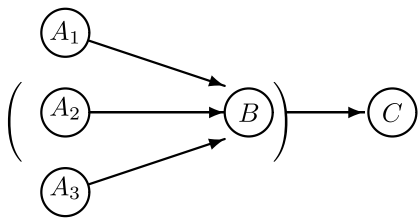


Figure 10.85: Family of networks

- Ω is a set. $\triangledown$

If the network has only one input node d, for simplicity, we can write just $\mathcal{N} = (S, R, d, E, \tau, V, L, \mathbb{F}, \Omega)$. Similarly, with respect to output nodes.

The elements of R represent connections between nodes. We perceive the coloring to propagate along the network using the arrows, the labels and the function $\mathbb{F}$. If the network has cycles, we expect V to be implicitly defined by $\mathbb{F}$.

Example 10.6.2 The network $\mathcal{N} = (S, R, d, e, \tau, V, L, \mathbb{F}, \Omega)$ where, in particular,

- $S = \{d, s, t, r, e\}$;
- $R = \{(d, r), (s, r), (t, r), (t, e), (r, e), (r, r)\}$;
- $L = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7\}$;
- $\tau(d, r) = l_1,\ \tau(d, s) = l_2,\ \tau(r, r) = l_3$
  $\tau(r, l) = l_4,\ \tau(s, r) = l_5,\ \tau(r, t) = l_6,\ \tau(t, l) = l_7$;

is depicted in Figure 10.87. The node d is the input point and the node e is the output point. The elements of R are represented by arrows. This network can be interpreted in several ways.

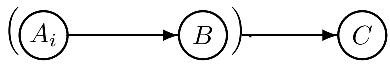


Figure 10.86: Network i

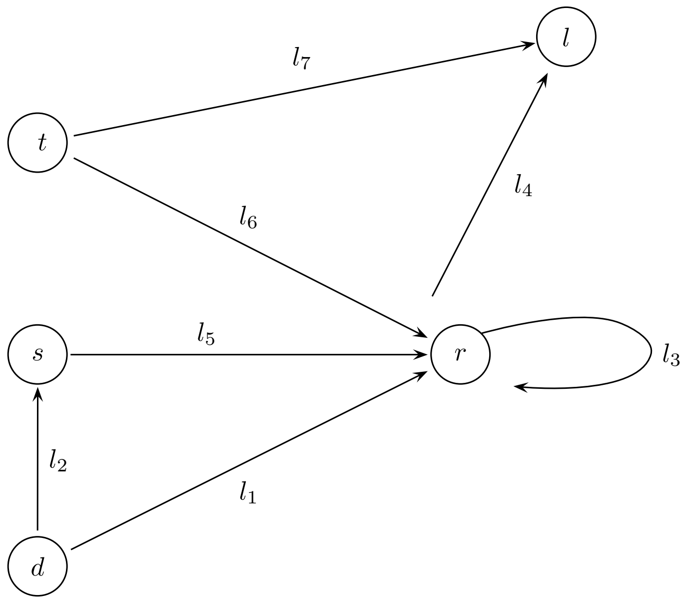


Figure 10.87: Example of network

(i) It can be interpreted as a map where the nodes are towns, the labels are distances and the colors are some heuristic numbers to aid some search function(for instance, the labels can give the aerial distance from a central point). We may require the graph to be acyclic. The function $\mathbb{F}$ can give the average distance of the parent nodes from the current node.

(ii) The network can be Bayesian, in which case we require it to be acyclic. We also require any point $t \neq d$ to either have a parent distinct from d (that is, for some x, $xRt, x \neq d$) or to have d alone as a parent. We forbid d itself to have parents.

Thus d is a dummy point ($d = \top$) showing the nodes without parents in the rest of the network. The function $\mathbb{F}$ would be the conditional probabilities of a node on its parents.

(iii) The network can be a neural net with τ and V different weights on the nodes and connections, and $\mathbb{F}$ some meaningful averaging function.

(iv) The network can be describing a flow problem with τ giving capacities, V giving retention and $\mathbb{F}$ is the obvious function summing up the flow. $\triangledown$

We now define the notion of fibring function.

Definition 10.6.3 Let $\mathbb{F}$ be a propagation family for states S and labels L with values in the space Ω. A function $\mathbf{F}$ giving a new propagation function for any triple $(V, l_i, \mathbb{F})$, $i = 1, \ldots, n$, is a *fibring function.* We write $\mathbf{F}$ as

$$\mathbf{F} : (V, l_i, \mathbb{F}) \mapsto \mathbb{F}_{V,l_i}$$

so $\mathbf{F}$ is defined for any set of labels. $\triangledown$

Note that in the above definition n is variable.

We now need to make some distinctions about fibring of network within networks. We give some additional examples.

Example 10.6.4 This is an example of refinement. Consider a network as depicted in Figure 10.88. Assuming that it is a map, d may be Durham and e may

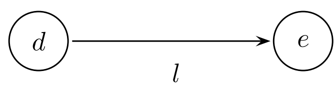


Figure 10.88: Map

be Edinburgh. Suppose that the label l is the number of heavy trucks per day one can push through from d to e. We can try and define this map by putting in for e another network, say, E which is the map on Edinburgh. This is substituting the actual sorting networks in the UK.

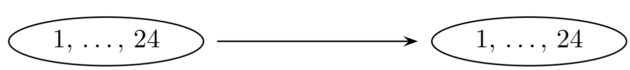


Figure 10.89: Refining days into hours

Another simpler example is when d and e are days and we can refine them into hours, as in Figure 10.89. $\triangledown$

Example 10.6.5 We get fibring/substitution of networks when we consider versions of the cut rule in labeled deductive systems. We describe a simple case.

Assume that our data is a list of formulas and the language contains $\rightarrow$ only. Thus, for example, we may have the list in Figure 10.90. We can perform modus ponens between any $X \rightarrow Y$ and X, provided X is immediately to its right and the result Y *replaces* $(X \rightarrow Y, X)$ in the list.

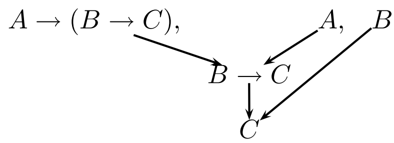


Figure 10.90: List of formulas

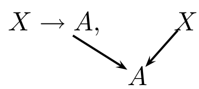


Figure 10.91: Proof of A

This way of doing modus ponens characterizes the one arrow *Lambek Calculus*. How would cut work? Suppose we have a proof of A in Figure 10.91.

We can simply substitute the sequence or net for A to get the list in Figure 10.92.

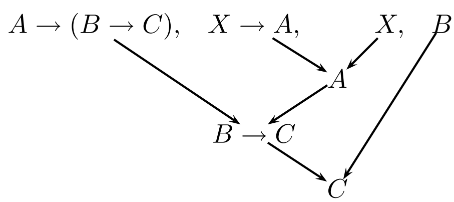


Figure 10.92: List after substitution

Suppose now that $X \to Y$ means a version of strict implication: if X holds next day then Y holds next day. The sequence

$$A \to (B \to C), A, B$$

can still be reduced to C but we must keep count of the days. Consider

$$C \to E,\ A \to (B \to C),\ A,\ B.$$

We can prove E in the Lambek Calculus (see Figure 10.93).

This will not work in the modal strict implication meaning of $\to$ because we must follow Figure 10.94. $C \to E$ is 3 days away from C. We need something like

$$\top \to (\top \to (C \to E)).$$

Thus, the network substitution of $(X \to A, X)$ into $(A \to (B \to C), A, B)$ should be different in the strict implication case. It should give the result in Figure 10.95.

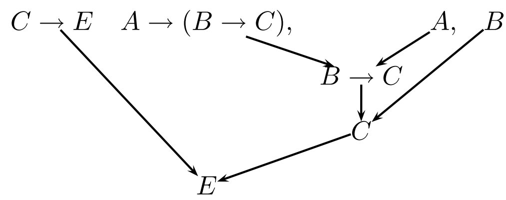


Figure 10.93: Proof of E

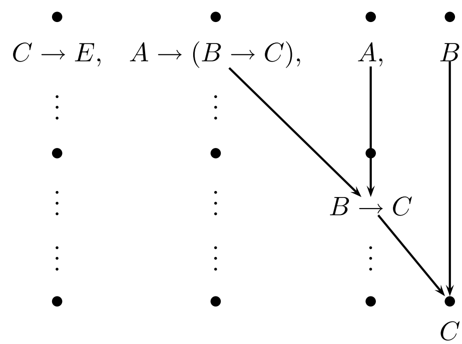


Figure 10.94: Modal strict implication

To summarize: $(X \to Y, X)$ is replaced by (Y) in the Lambek Calculus and is replaced by $(\cdot, Y)$ in the strict implication logic. Thus, the network substitutions corresponding to these logics are as described bellow.

If

$$N_1 = (x_1, \ldots, x_n, y, z_1, \ldots, z_m)$$
$$N_2 = (u_1, \ldots, u_k)$$

then the *Lambek substitution* is

$$N_1(y/N_2) = (x_1, \ldots, x_n, u_1, \ldots, u_k, z_1, \ldots, z_m).$$

and the *Strict substitution* is

$$N_1(y/N_2) = (x_1 \wedge u_1, \ldots, x_n \wedge u_n, u_{n+1}, z_1, \ldots, z_m)$$

when $k = n+1$. If $k \neq n$, add $\top$'s to the beginning of the shorter one to make them equal and then substitute.

From this example we conclude that if the networks represent some logic, then options for fibring networks represent options for the cut rule in the logic. ∇

We need to prepare the ground for the general definition of fibring which will follow. So, consider the networks $\mathcal{N}_1$ and $\mathcal{N}_2$ in Figures 10.96 and 10.97, respectively.

$$\begin{array}{ccc} \bullet & \bullet & \bullet \\ A \to (B \to C) & X & B \\ X \to A & & \end{array}$$

Figure 10.95: Network after substitution

There are have several options in substituting the network $\mathcal{N}_2$ for t^1, using a fibring function $\mathbf{F}$.

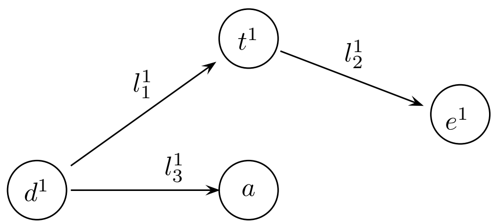


Figure 10.96: Network $\mathcal{N}_1$

The most straightforward one is to replace t^1 by $\mathcal{N}_2$ and redirect all arrows coming into t^1 and connect them to all input points d_j^2 of $\mathcal{N}_2$. Similarly, all arrows

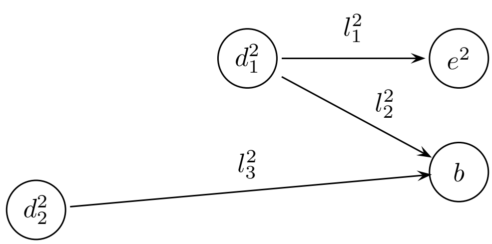


Figure 10.97: Network $\mathcal{N}_2$

coming out of t^1 will now come out of the output point of $\mathcal{N}_2$. The resulting network is depicted in Figure 10.98.

The function $\mathbb{F}^{1,2}$ is the same as $\mathbb{F}^1$ on nodes from $\mathcal{N}_1$ and is the fibred function $\mathbb{F}^2_{V^1_{(t^1)}, l_1^1, l_2^1}$ obtained by applying $\mathbf{F}$ to $\mathbb{F}^2$.

Variations can be obtained by changing $\mathbb{F}$ and/or by changing the input output points of $\mathcal{N}_2$ before fibring. So this is quite a general definition. The basic idea is that the "environment" of t^1 (namely , V^1/t^1 and all labels of connections leading into and out of t^1) change the fibring function $\mathbb{F}^2$ of the substituted network $\mathcal{N}_2$ into $\mathbf{F}(\mathbb{F}^2)$.

Problems may arise if either $\mathcal{N}_1$ and $\mathcal{N}_2$ have nodes in common or t^1 is connected to itself. This can cause more than one arrow to occur between two points. For

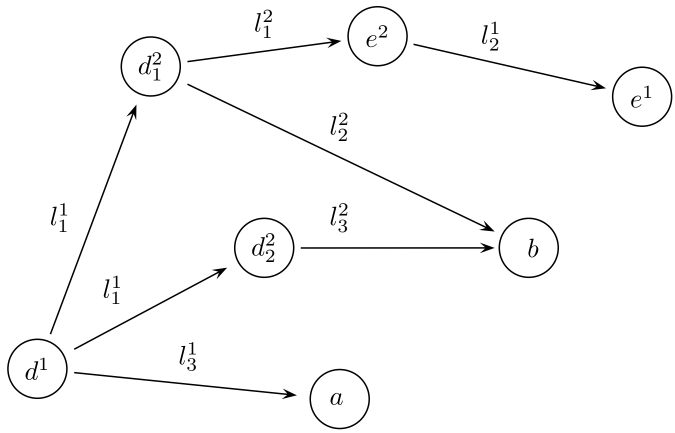


Figure 10.98: Network obtained after replacing t^1 by $\mathcal{N}_2$

this reason these situations are excluded. To see why this can happen, imagine we substitute the network $\mathcal{N}_3$ depicted in Figure 10.99 into the network $\mathcal{N}_4$ depicted in

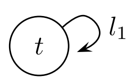


Figure 10.99: Network $\mathcal{N}_3$

Figure 10.100, where t is both the input and output points. We get, by definition,

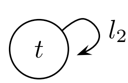


Figure 10.100: Network $\mathcal{N}_4$

the network depicted in Figure 10.101.

We use the following notation: (i) if $\mathcal{N}_1$ and $\mathcal{N}_2$ are described using $\rightarrow$ and t^1 is in $\mathcal{N}_1$, we can indicate fibring by substituting $\mathcal{N}_2$ for t^1 and using $\twoheadrightarrow$ to connect into and out of $\mathcal{N}_2$ and (ii) the fibring function $\mathbf{F}$ is suppressed.

Example 10.6.6 We use the representation depicted in Figure 10.102 to indicate a connection between X and Y, where X is a parent of Y.

An example of network using $\rightarrow$ is depicted in Figure 10.103.

We use $\twoheadrightarrow$ as a special connection between a node X and a network $\mathcal{N}$. If $\mathcal{N}$ is the network in Figure 10.103 and $\twoheadrightarrow$ means that we connect with every node

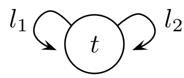


Figure 10.101: Network after substitution

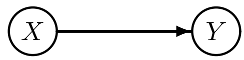


Figure 10.102: Example of connection

in the network using $\rightarrow$, then $X \twoheadrightarrow \mathcal{N}$ is the network depicted in Figure 10.104. Arrows coming out of $\mathcal{N}$ into Y are not drawn. $\triangledown$

We now conclude with a general definition:

Definition 10.6.7 Let $\mathcal{N}_i = (S^i, R^i, D^i, E^i, \tau^i, V^i, L^i, \mathbb{F}^i, \Omega^i)$ for $i = 1, 2$, be networks such that $S^i \subseteq S$, $L^i \subseteq L$ and $\Omega^i \subseteq \Omega$. Assume also that $S^1 \cap S^2 = \emptyset$. Let $\mathbf{F}$ be a fibring function, let $t^1 \in S^1$ be a node such that not $t^1 R^1 t^2$, and let l^1_j be all the labels of nodes in $\mathcal{N}_1$ leading into or coming out of t^1.

The *one step fibred system* $\mathcal{N}_1(t^1/\mathcal{N}_2)$ is the network

$$\mathcal{N}_{1,2} = \langle S^{1,2}, R^{1,2}, D^{1,2}, E^{1,2}, \tau^{1,2}, V^{1,2}, L^{1,2}, \mathbb{F}^{1,2}, \Omega^{1,2} \rangle$$

where

- $S^{1,2} = (S^1 \cup S^2) \backslash \{t^1\}$;
- $R^{1,2} = (R^1 \cup R^2 \cup \{(x, d^2_i) : xR^1t^1\} \cup \{(e^2_j, y) : t^1 R^1 y\}) \backslash \{(x, y) \in R^1 : x = t^1 \text{ or } y = t^1\}$;
- $D^{1,2} = \{d^1_i : t^1 \neq d^1_i\} \cup \{d^2_k : t^1 \text{ is an input point}\}$;
- $E^{1,2} = \{e^1_j : t^1 \neq e^1_j\} \cup \{e^2_k : t^1 \text{ is an output point}\}$;
- $\tau^{1,2}((x, y)) = \begin{cases} \tau^1(x, y) & \text{if } x, y \in S^1 \text{ and } xR^{1,2}y \\ \tau^2(x, y) & \text{if } xR^{1,2}y \text{ and } x, y \in S^2 \\ \tau^1(x, t^1) & \text{if } x \in S^1, y \in S^2 \text{ and } xR^{1,2}y \\ \tau^1(t^1, y) & \text{if } x \in S^2, y \in S^1 \text{ and } xR^{1,2}y \end{cases}$
- $\Omega^{1,2} = \Omega^1 \cup \Omega^2$;
- $L^{1,2} = L^1 \cup L^2$;

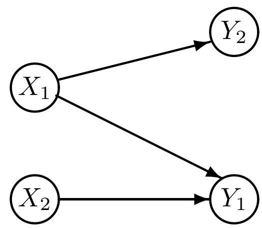


Figure 10.103: Example of network using $\to$

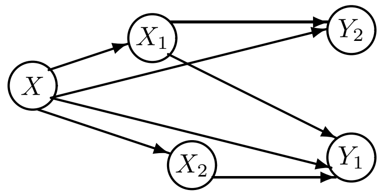


Figure 10.104: $X \twoheadrightarrow \mathcal{N}$

- $\mathbb{F}^{1,2}$ is defined as $\mathbf{F}(V^1(t^1), l_i^1, \mathbb{F}^2)$ where l_i are all the labels from other nodes in S^1 leading to t^1 and labels of nodes in S^1 into which t^1 leads. We assume $\mathbf{F}$ is such that $\mathbb{F}^{1,2} = \mathbb{F}^1$ on points in S^1 (this is possible since we assumed $S^1 \cap S^2 = \emptyset$). ∇

10.7 Final remarks

In this chapter, we wanted to illustrate that the concept of fibring is useful in contexts that do not explicitly involve logics. The examples range from classical computation to neural nets and Bayesian networks. The basic idea is to model components by networks seen as labeled deductive systems. The fibring techniques are useful when we want to be able to replace nodes in a network by another network. The advantage of using labeled deductive systems is that we bring to these fields logical tools. A general notion that accommodates all the case studies is self-fibring. It is still to be understood the kind of properties that we may want to preserve when we deal with self-fibring of networks.

We go on identifying some aspects to be further pursued for the case studies. Moreover, we will compare several issues discussed in this chapter with related work.

We start with neural networks. The question of which logics could be represented in fibred neural networks (fNNs) is an interesting open question. The natural next step is to use the recursive, more expressive architecture of fNNs to perform symbolic computation, giving fNNs a neural-symbolic characterization.

It is expected that fNNs can be used to represent variables, as well as to learn and reason about relational knowledge. Another interesting work to pursue is to define fibring of recurrent neural networks. Finally, the questions of how different networks should be fibred and which fibring functions should be used are very important ones when it comes to practical applications of fNNs. This is clearly domain dependent, and an empirical evaluation of fNNs in comparison with standard neural networks would be required.

Then we turn our attention to Bayesian networks. We started by summarizing recursive Bayesian multinets along the lines of [219]. These nets generalize Bayesian networks. They represent context-specific independence relationships by a set of Bayesian networks, each of which represents the conditional independencies which operate in a fixed context. By creating a variable C whose assignments yield different contexts, a Bayesian multinet may be represented by a decision tree whose root is C and whose leaves are the Bayesian networks. The idea behind recursive Bayesian multinets is to extend the depth of such decision trees. Root nodes are still Bayesian networks, but there may be several decision nodes. Recursive Bayesian multinets are rather different from recursive Bayesian networks we have considered: they are applicable to context-specific causality where the contexts need to be described by multiple variables, not to general instances of recursive causality, and consequently they are structurally different, being decision trees whose leaves are Bayesian networks rather than Bayesian networks whose nodes take Bayesian networks as values.

Recursive relational Bayesian networks generalize the expressive power of the domain over which Bayesian networks are defined [155]. Bayesian networks are essentially propositional in the sense that they are defined on variables, and the assignment of a value to a variable can be thought of as a proposition which is true if the assignment holds and false otherwise. We have made this explicit by representing the two possible assignments to variable A by a and $\neg a$ respectively. Relational Bayesian networks generalize Bayesian networks by enabling them to represent probability distributions over more fine-grained linguistic structures. Recursive relational Bayesian networks generalize further by allowing more complex probabilistic constraints to operate, and by allowing the probability of an atom that instantiates a node to depend recursively on other instantiations as well as the node's parents (see [155] for the details). Thus, in the transition from relational Bayesian networks to recursive relational Bayesian networks the Markovian property of a node being dependent just on its parents (not further non-descendants) is lost. Therefore, recursive relational Bayesian networks and recursive Bayesian networks differ fundamentally with respect to both motivating applications and properties.

Object-oriented Bayesian networks were developed as a technique for representing large-scale Bayesian networks efficiently [163]. Object-oriented Bayesian networks are defined over objects (of which a variable is an example). Such networks are very general in principle.

Recursive Bayesian networks are instances of object-oriented Bayesian networks in as much as recursive Bayesian networks can be formulated as objects in the object-oriented programming sense. Moreover, in practice, object-oriented Bayesian networks often look much like recursive Bayesian networks, in that such a network may contain several Bayesian networks as nodes, each of which contains further Bayesian networks as nodes and so on (see [216] for example). However, there is an important difference between the semantics of such object-oriented Bayesian networks and that of recursive Bayesian networks, and this difference is dictated by their motivating applications.

Object-oriented Bayesian networks tend to be used to organize information contained in several Bayesian networks: each such Bayesian network is viewed as a single object node in order to hide much of its information that is not relevant to computations being carried out in the containing network. So by expanding each Bayesian network node, an object-oriented Bayesian network can be expanded into one single non-recursive, non-object-oriented Bayesian network. In contrast, in a recursive Bayesian network, recursive Bayesian networks occur as *values* of nodes not as nodes themselves, and when one recursive Bayesian network b_1 causes another b_2 in a containing recursive Bayesian network b, it is not output variables of b_1 that cause input variables of b_2, it is b_1 *as a whole* that causes b_2 *as a whole*. Correspondingly, there is no straightforward mapping of a recursive Bayesian network on V to a Bayesian network on V. Thus, while object-oriented Bayesian networks are in principle very general, in practice they are often used to represent very large Bayesian networks more compactly by reducing sub-networks into single nodes. In such cases the arrows between nodes in an object-oriented Bayesian network are interpreted very differently to arrows between nodes in a recursive Bayesian network, and issues such as causal, Markov and probabilistic consistency do not arise in the former formalism.

Finally, hierarchical Bayesian networks (HBNs) were developed as a way to allow nodes in a Bayesian network to contain arbitrary lower-level structure [139]. Thus, recursive Bayesian networks can be viewed as one kind of HBN, in which lower-level structures are of the same type as higher-level structures, namely Bayesian network structures. However, there are a number of important differences. HBNs seem to have been developed in order to achieve extra generality, while recursive Bayesian networks were created in order to model an important class of causal claims. HBNs have been developed in most detail, namely in the case where lower-level structure corresponds to causal connections. However, the lower-level structures are not exactly Bayesian networks in HBNs. Indeed, one must specify the probability of each variable conditional on its parents in its local graph and all variables higher up the hierarchy. Thus, HBNs have much larger size complexity than recursive Bayesian networks. HBNs do not adopt the recursive Markov condition mentioned in Section 10.5. They only assume that a variable is probabilistically independent of all nodes that are not its descendants conditional on its parents and all higher-level variables. This has its advantages and its disadvantages. On the one hand, it is a weaker assumption and thus less open to question. On the other hand, it leads to the larger size of HBNs and in any case the recursive Markov condition is rather plausible. Finally, variables can only appear once in a HBN, but they can appear more than once in a recursive Bayesian network. Thus, HBNs are more restrictive than recursive Bayesian networks in one respect, and more general in another, and have quite different probabilistic structure. However, they share common ground too, and where one approach is inappropriate, the other might well be applicable.

Chapter 11

Summing-up and outlook

In this chapter we start by presenting a brief synthesis of the main features and properties of the combination mechanisms discussed in this book. Then, we point out some applications where we believe that the techniques proposed so far can be useful. We emphasize fibring techniques and results that can be useful there. We also review other combination mechanisms that appeared motivated by software engineering applications. Afterward, we discuss emergent applications that need, besides the existing techniques, other forms of combination and even new logic connectives. Finally, we indicate new directions of research not only in fibring but also related to new combination mechanisms.

The structure of this chapter is as follows. In Section 11.1, we review the main features of the combination mechanisms discussed in this book. We also refer to preservation results. In Section 11.2, we present an example of the application of fibring to knowledge representation endowed with a deontic component. We also point out the interest of combination techniques in the field of distributed systems. In Section 11.3, we show how network fibring can be applied to the area of argumentation theory. In Section 11.4, we refer to the application of the fibring concepts to specification and verification of reactive systems. We briefly describe specific combination mechanisms for software engineering such as parameterization, temporalization and synchronization. We also refer to the use of fibring techniques in the context of institutions and parchments. In Section 11.5, we point out some emergent and potential applications, like security, quantum computation and space-time systems. Finally, in Section 11.6, we give some hints on research directions in fibring and point out new forms of combination like probabilization and quantization.

11.1 Synthesis

We start with a synthesis of the main features of the combination mechanisms that we discussed in this book, as well as the preservation results that we have for each of them.

In Figure 11.1, we present the main features of the combination mechanisms. We use the prefixes **d** and **s** for denoting the deductive and the semantic components of a given combination mechanism, respectively. We observe that the fibring lines include all the settings where we studied fibring. The line **h-fibring1** refers to heterogeneous fibring via consequence systems and line **h-fibring2** refers to heterogeneous fibring via abstract deductive systems.

	heterogeneous	**pre-process**	**algorithmic**	**minimal**
d-fusion	**N**	**N**	**Y**	**Y**
s-fusion	**N**	**N**	**Y**	**Y**
d-product	**N**	**N**	**N**	**Y**
s-product	**N**	**N**	**Y**	**N**
d-fibring	**N**	**N**	**Y**	**Y**
s-fibring	**Y**	$\mathbf{Y_b}$	**Y**	**Y**
h-fibring1	**Y**	$\mathbf{Y_a}$	**Y**	**Y**
h-fibring2	**Y**	$\mathbf{Y_b}$	**Y**	**Y**

Figure 11.1: Features of combination mechanisms

The **heterogeneous** column states that the mechanism can be applied to logics described in a different way.

The **pre-process** column indicates whether some processing has to be done on the original logics before the combination itself. For instance, the fibring, in many chapters of the book, assumes that both logics are presented by Hilbert calculi and no pre-processing is needed before combining them. We indicate that no processing is needed by **N**. Otherwise, we distinguish two situations: either the mechanism states how to do the pre-processing, indicated by $\mathbf{Y_a}$, or it does not, indicated by $\mathbf{Y_b}$, and one has to figure out how to do it. In Chapter 3, given a logic with no algebraic semantics, one has to say how the semantic structures induce ordered algebras. On the other hand, in Chapter 4, two cases have to be considered. In the case of fibring consequence systems, the pre-processing step is immediate, since we only consider logics that induce consequence systems. In the case of fibring abstract deductive systems, one has to say how a given logic induces an abstract deductive system. We have already described how to do so for many kinds of calculi (such as Hilbert calculus, tableau calculus, most sequent calculus and natural deduction calculus).

The **algorithmic** column indicates that the resulting logic is completely defined by the combination mechanism. Finally, the **minimal** column states that the constructions yields a minimal logic (in some class of logics). We use **Y** and **N** to indicate whether or not a combination mechanism has this feature.

Finally, we observe that all the mechanisms discussed so far are symmetric in the sense that the order in which we take the original logics is not relevant.

We turn now our attention to preservation results. In Figure 11.2, we refer to the properties of preservation of the metatheorem of deduction (MTD), interpolation and congruence that were mentioned in this book.

	MTD	interpolation	congruence
d-fusion	**Y**		**Y**
s-fusion		**Y**	
d-fibring	**Y**	**Y**	$\mathbf{Y_{sc}}$

Figure 11.2: Preservation of properties

The empty boxes mean that the property was not discussed for that mechanism. As pointed out in Chapter 2, congruence is not always preserved by fibring. But we proved a sufficient condition for the preservation of congruence. We indicate this fact by $\mathbf{Y_{sc}}$.

We can go further and investigate the properties of the operator $\odot$ underlying each combination mechanism. When $\odot$ is either fusion or fibring we have:

- $\mathcal{L}' \odot \mathcal{L}'' = \mathcal{L}'' \odot \mathcal{L}'$ commutativity
- $(\mathcal{L}' \odot \mathcal{L}'') \odot \mathcal{L}''' = \mathcal{L}' \odot (\mathcal{L}'' \odot \mathcal{L}''')$ associativity
- $\mathcal{L} \odot \mathcal{L} = \mathcal{L}$ idempotence

The reader should note that there is just one exception to idempotence in fibring (recall Section 5.5 of Chapter 5).

With respect to the deductive component of fibring, these properties are straightforward. With respect to the semantic component of fibring it is also easy to see that this properties hold, namely because the carrier sets of the algebras in the fibring are such that:

$$\{B' : \langle B', \leq', \nu' \rangle \in \mathcal{A}'\} \cap \{B'' : \langle B'', \leq'', \nu'' \rangle \in \mathcal{A}''\}$$

where $\mathcal{A}'$ and $\mathcal{A}''$ are the classes of ordered algebras in $\mathcal{L}'$ and $\mathcal{L}''$, respectively. The case of fusion is similar.

In Figure 11.3, we refer to the properties of preservation of soundness, completeness and weak completeness (w-completeness). In the case of fibring, as discussed in Chapter 3, we do not prove directly that completeness is preserved. Instead, we prove a general completeness theorem for a class of logics, including some sufficient conditions. Then we prove that the sufficient conditions are preserved by fibring. In the case of fusion, preservation of weak completeness can be directly proved.

There are two typical general scenarios where the existing combination mechanisms can be used. The first one is when one wants to join logics that are already known. That is, assume that the logics are endowed with a Hilbert calculus and

	soundness	**completeness**	**w-completeness**
fusion	**Y**		**Y**
fibring	**Y**	$\mathbf{Y_{sc}}$	

Figure 11.3: Preservation of properties

semantics. For instance, someone wants to have a logic expressing knowledge and normative assertions. Moreover, assume that we know which modal logic we want for knowledge and which deontic logic we want for obligation. In this case, we know how to do the combination using the mechanism of fibring.

In general, we have an algorithmic solution for defining the fibring $\mathcal{L}$ of the logics $\mathcal{L}'$ and $\mathcal{L}''$ with signatures C' and C'', Hilbert calculi H' and H'' and classes M' and M'' of semantic structures, respectively:

- obtain the signature C as the fibring of C' and C'' as in Section 2.1 of Chapter 2;
- obtain the Hilbert calculus H as the fibring of the given Hilbert calculi H' and H'' as described in Section 2.2 of Chapter 2;
- get the induced algebraic semantics, as explained in Chapter 3, from the given semantics of $\mathcal{L}'$ and $\mathcal{L}''$ and check that semantic consequence is preserved;
- obtain the algebraic fibring of the induced algebraic structures as defined in Section 3.1 of Chapter 3;
- the resulting logic $\mathcal{L}$ is sound providing that the original logics $\mathcal{L}'$ and $\mathcal{L}''$ are sound as proved in Section 3.3 of Chapter 3;
- check that the sufficient conditions for completeness are verified for $\mathcal{L}'$ and $\mathcal{L}''$ and use the preservation result to conclude that the resulting logic is complete, as in Section 3.3 of Chapter 3;
- if some interaction axioms are needed, join them to the resulting Hilbert calculus and restrict the semantic algebraic structures to the ones that satisfy the new axioms (soundness and completeness still hold).

In the above, we are assuming propositional based logics $\mathcal{L}'$ and $\mathcal{L}''$, for simplification. Clearly, algorithmic constructions similar to the above can be set up for first-order, higher-order and some non-truth functional logics according to the corresponding chapters.

Let us turn now our attention to the second general scenario where combination mechanisms can be used. In this second situation, we start with a logic with several connectives, for example, space and time connectives, and we want to study some of its properties such as, for instance, completeness. Then, we may start by isolating, on one hand, the space component and, on the other hand, the time component, in such a way that the fibring of these two components is the original logic. Afterward, we check that the properties hold for the space and the time components and use the fibring preservation results to conclude that the original logic has the desired properties.

11.2 Knowledge representation and agent modeling

It seems that fibring can be directly applicable to many situations in artificial intelligence in general and knowledge representation in particular. Also in the field of distributed systems (and agent modeling) fibring seems to have a role.

Normative knowledge representation

Assume that we want to put together in the same logic a modal operator $\Box'$ for dealing with obligation and another modal operator $\Box''$ for expressing knowledge. In other words, we want to combine a deontic logic with an epistemic logic.

We may assume that the deontic logic and the epistemic logic are respectively the non-normal modal logic $\mathcal{L}'$ (see [54]) and the normal modal $\mathcal{L}''$ ([149], [91]) described below. Propositional connectives are shared. In the sequel, $\bot$ is an abbreviation of $(\neg(\xi_1 \Rightarrow \xi_1))$.

- Hilbert calculi H' and H'' include besides the propositional axioms and modus ponens the following axioms and rules:
  - $\langle\emptyset, (\neg(\Box'\bot))\rangle$;
  - $\langle\{(\xi_1 \Rightarrow \xi_2)\}, ((\Box'\xi_1) \Rightarrow (\Box'\xi_2))\rangle$ (global rule);
  - $\langle\emptyset, ((\Box''(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box''\xi_1) \Rightarrow (\Box''\xi_2)))\rangle$;
  - $\langle\emptyset, ((\Box''\xi_1) \Rightarrow \xi_1)\rangle$;
  - $\langle\emptyset, ((\Box''\xi_1) \Rightarrow (\Box''(\Box''\xi_1)))\rangle$;
  - $\langle\emptyset, ((\neg(\Box''\xi_1)) \Rightarrow ((\Box''(\neg(\Box''\xi_1)))))\rangle$;
  - $\langle\{\xi_1\}, (\Box''\xi_1)\rangle$ (global rule).
- Semantics consists of classes of Kripke structures M' and M'':
  - M' consists of all structures of the form

$$\langle W', N', V'\rangle$$

    such that W' is a non-empty set, $N' : W \to \wp(\wp W)$ is a map and $V' : \Pi \to \wp W$ such that $\emptyset \notin N(w)$ for every $w \in W$;

- M'' consists of all the usual Kripke structures

$$\langle W'', R'', V'' \rangle$$

where R'' is an equivalence relation;

- Among the interaction axioms we may want to have:

$$\langle \emptyset, ((\Box'(\Box''\xi_1)) \Rightarrow (\Box''\xi_1)) \rangle$$

stating that if it is obligatory to know p then one should know p.

The satisfaction for $\Box'$, in the logic $\mathcal{L}'$, is as follows:

$$\langle W', N', V' \rangle, w' \Vdash' (\Box'\varphi') \quad \text{if} \quad \{u' \in W' : \langle W', N', V' \rangle, u' \Vdash' \varphi'\} \in N(w').$$

The satisfaction for the other connectives is the usual one.

Given a structure $\langle W', N', V' \rangle$ for the deontic logic, the induced ordered algebra $\langle B', \leq', \nu' \rangle$ is defined as follows:

- B' is $\wp W'$;
- $b'_1 \leq b'_2$ if $b'_1 \subseteq b'_2$;
- $\nu'(\Box')(b') = \{w' \in W' : b' \in N'(w')\}$;
- $\nu'(\neg)(b') = W' \setminus b'$;
- $\nu'(\Rightarrow)(b'_1, b'_2) = (W' \setminus b'_1) \cup b'_2$.

The ordered algebra $\langle B'', \leq'', \nu'' \rangle$ induced by the Kripke structure $\langle W'', R'', V'' \rangle$ for the epistemic logic is as defined in Chapter 3. Note that in the algebra $\langle B', \leq', \nu' \rangle$, the only difference to what was discussed in that chapter is the definition of $\nu'(\Box')$.

Then the fibring

$$\mathcal{L} = \mathcal{L}' \cup \mathcal{L}''$$

is as follows.

- Hilbert calculus H includes all the axioms and rules of H' and H'', that is, it include besides the propositional axioms and modus ponens the following axioms and rules:
  - $\langle \emptyset, (\neg(\Box'\bot)) \rangle$;
  - $\langle \{(\xi_1 \Rightarrow \xi_2)\}, ((\Box'\xi_1) \Rightarrow (\Box'\xi_2)) \rangle$;
  - $\langle \emptyset, ((\Box''(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box''\xi_1) \Rightarrow (\Box''\xi_2))) \rangle$;
  - $\langle \emptyset, ((\Box''\xi_1) \Rightarrow \xi_1) \rangle$;
  - $\langle \emptyset, ((\Box''\xi_1) \Rightarrow (\Box''(\Box''\xi_1))) \rangle$;

- $\langle \emptyset, ((\neg(\Box''\xi_1)) \Rightarrow ((\Box''(\neg(\Box''\xi_1)))))\rangle$;
- $\langle \{\xi_1\}, (\Box''\xi_1)\rangle$.

- Semantics consists of the class of all ordered algebras

$$\langle B, \leq, \nu\rangle$$

such that:

- $\langle B, \leq, \nu|_{\neg,\Rightarrow,\Box'}\rangle$ is the ordered algebra induced by some $m' \in M'$, where $\nu|_{\neg,\Rightarrow,\Box'}(c) = \nu(c)$ for every $c \in \{\neg, \Rightarrow, \Box'\}$;
- $\langle B, \leq, \nu|_{\neg,\Rightarrow,\Box''}\rangle$ is the ordered algebra induced by some $m'' \in M''$, where $\nu|_{\neg,\Rightarrow,\Box''}(c) = \nu(c)$ for every $c \in \{\neg, \Rightarrow, \Box''\}$.

The logic $\mathcal{L}^+$ corresponding to adding the interaction axiom referred to above is endowed with the Hilbert calculus H^+ which is H plus the interaction axiom. The semantic is provided by all the ordered algebras for $\mathcal{L}$ that satisfy the interaction axiom. In the logic $\mathcal{L}^+$, we can make reasoning about knowledge and obligation (freely mixing the obligation operator $\Box'$ and the knowledge operator $\Box''$). Moreover, we can import some properties of this logic for free.

We start by observing that the logic $\mathcal{L}^+$ is sound. This conclusion is simply achieved in two steps. First, using the preservation for soundness, we obtain the soundness of $\mathcal{L}$ since $\mathcal{L}'$ and $\mathcal{L}''$ are sound. Secondly, choosing the models that satisfy the interaction axiom we ensure the soundness of $\mathcal{L}^+$.

With respect to completeness of $\mathcal{L}$ we have to enlarge the class of ordered algebras for $\mathcal{L}'$ and $\mathcal{L}''$ with all the ordered algebras that satisfy the axioms and rules of $\mathcal{L}'$ and $\mathcal{L}''$ (not just the induced ones). Note that this enrichment does not affect soundness. Then, applying the result of preservation of completeness we conclude that $\mathcal{L}$ is complete (with respect to this enlarged semantics). By choosing all the ordered algebras in the enlarged class that also satisfy the interaction axiom, we get the completeness of $\mathcal{L}^+$ by using the general completeness result.

Distributed systems

We now briefly point out that fibring can also be useful in the field of distributed systems and agent modeling.

The typical situation is to have n systems or agents that have individual knowledge and can interact with each other. Moreover, in some cases, it is also important to investigate properties of a collection of those systems.

We are interested in the cases where the problem is studied in a logical setting. That is, each system i is defined in a particular (local) logic $\mathcal{L}_i$, that is usually the same, interaction is expressed in a global logic and properties about collections of systems can be stated using new logical operators.

Among the approaches we can refer to the one presented in [91]. Therein, each system i is described by a modal logic with a unary modal operator $\Box_i$, for instance as a $S5$ modal operator. In distributed systems and agent modeling, it is expected that we want some interaction axioms. For instance saying that if system i knows α then we want system j also to know α. In order to be able to state this kind of assertions we can define the fibring $\mathcal{L}$ of $\mathcal{L}_1, \ldots, \mathcal{L}_n$. As a result we obtain a language where we can mix connectives that belong to the signatures of the different systems. Then, we can add interactions through axioms in the language of $\mathcal{L}$. For the example above we may add the following axiom:

$$((\Box_i \alpha) \Rightarrow (\Box_j \alpha)).$$

Moreover, we may also add axioms like

$$((\Box_i (\Box_j \beta)) \Rightarrow (\Box_i \beta))$$

meaning that if system i knows that system j knows β then system i also knows β.

In this way, all the fibring techniques that we summarized in the beginning of this chapter can be applied. In particular, preservation of metaproperties can be immediately applied.

Another advantage of using fibring techniques in this context is that we can explore situations where the systems or agents are not necessarily described by the same logic.

Another dimension, showing that fibring is relevant is this context, is related to the so called common knowledge applications. In this kind of applications, we can have new operators E_G and C_G where G is a collection of distributed systems. The formula

$$((E_G \varphi) \Leftrightarrow (\bigwedge_{i \in G} (\Box_i \varphi)))$$

meaning that everyone in collection G knows φ, and the formula

$$((C_G \varphi) \Leftrightarrow (E_G(\varphi \wedge (C_G \varphi)))$$

meaning that φ is common knowledge among the systems in G, are taken as axioms. Both formulas can only be written in the language of fibring.

Another approach to distributed systems is based on event structures [277]. Herein we follow [86]. In this case, all the systems are described using the same temporal logic. The interactions are described by labeled formulas of the type $i : \varphi$ where the label i represents system i. The formula φ may involve labeled formulas as well. For instance the formula

$$i.(\varphi \Rightarrow j.(X\psi))$$

means that the following holds for system i: if φ holds then, for system j, formula ψ holds in the next state. At the semantic level, event structures are used.

We believe that fibring techniques can be applied successfully at the deductive level. In order to take advantage of the fibring techniques for preservation of metaproperties involving also semantic aspects some research has to be done.

There are two possibilities. Either we can study how event structures can induce ordered algebras or we can investigate what is the fibring of event structures themselves.

This is a topic worthwhile to be pursued since event structures with a probability dimension are currently under study (see [269]).

11.3 Argumentation theory

Argumentation theory is the science of effective civil debate or dialogue, using rules of inference and logic. Argumentation is concerned primarily with reaching conclusions through logical reasoning based on certain premises. Although including debate and negotiation which are concerned with reaching mutually acceptable conclusions, argumentation theory also encompasses the branch of social debate.

Typically, an argument has an internal structure, comprising

- a set of assumptions or premises;
- a method of reasoning or deduction;
- a conclusion or point.

Classical logic is often used as the method of reasoning so that the conclusion follows logically from the assumptions. However, if the set of assumptions is inconsistent then anything can logically follow from inconsistency. Therefore, it is common to insist that the set of assumptions is consistent. It is also good practice to require the set of assumptions to be the minimal set, with respect to set inclusion, necessary to infer the consequent. Such argumentation has been applied to the fields of law and medicine. A second school of argumentation investigates abstract arguments that by definition have no internal structure.

Abstract argumentation systems were put forward as a response to the realization that no argument or proof is conclusive in real life, and that arguments have counterarguments. An argument framework is a pair

$$AF = (AR, Attacks)$$

where

- AR is a set of objects called arguments;
- $Attacks$ is a binary relation (usually irreflexive), saying which arguments x attacks which argument y.

An example of argument framework is depicted in Figure 11.4. In this framework, a attacks c, c attacks b and b attacks a. There are no winning arguments here.

This framework is too abstract to be of specific use. It equally applies to circuits and impending circuits, credits and debits, or any system involving x and anti-x,

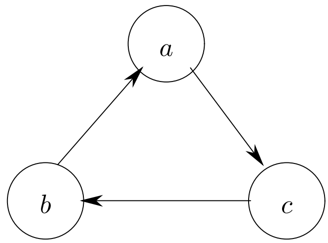


Figure 11.4: Example of argumentation framework

whatever x is. To apply such a system successfully we need to go into the structure of the arguments and analyze the mechanics of one argument attacking another.

An improvement was proposed in [19] by introducing value-based argumentation frameworks. In such a framework we are given a set of colors (values) and a coloring of the arguments. The values are partially ordered and an argument of strictly lesser value cannot now attack an argument of stronger value. If, for example, we make b red and a and c blue in the previous figure then

- if blue is stronger than red, then b cannot attack and defeat a, a can attack c and the winning arguments are $\{a, b\}$, because c is out;
- if red is stronger than blue then the winning arguments are $\{b, c\}$.

Although an interesting improvement, this model is still too abstract. Real life has arguments within arguments in different levels and interconnections between the levels. The mechanism of self-fibring of networks referred to in Chapter 10 can be used to extend this approach.

Labeled deductive systems can also be used. For example, in the context of LDS, the above situation will arise if we have a labeled database which includes items such as

$$t : a, \quad s : b, \quad r : c$$

and some additional data, say $u_i : X_i$, such that the following can be proved, among others

- $\gamma(t) : (\neg c)$;
- $\beta(r) : (\neg b)$;
- $\alpha(s) : (\neg a)$;

where α, β, γ are the labels of $(\neg a)$, $(\neg b)$ and $(\neg c)$, respectively. In these labels t, r and s are mentioned to indicate that, for instance, $t : a$ is used in the proof of $\gamma(t) : (\neg c)$. Hence, a with label t attacks c, by proving $(\neg c)$ with label $\gamma(t)$. The label $\gamma(t)$ shows exactly what role a plays in this attack. We are assuming that to defeat x we must put forward an argument for $(\neg x)$. This is only a simplifying assumption. In LDS, x comes with a label t and so to weaken $t : x$ we can attack t.

The flattening process acts here as value judgment of what can win, $r : c$ or $\gamma(t) : (\neg c)$, by comparing r and $\gamma(t)$. Obviously, the value based argumentation machinery can be used as part of the flattening process.

The following LDS model reflects the colored diagram mentioned above:

$$\begin{aligned} \text{red: } & b \\ \text{blue: } & a \\ \text{blue: } & c \\ \text{red to blue: } & b \to (\neg a) \\ \text{blue to blue: } & a \to (\neg c) \\ \text{blue to red: } & c \to (\neg b) \end{aligned}$$

Using modus ponens in the form

$$\frac{\alpha : X, \beta : X \to Y, \varphi(\beta, \alpha)}{\alpha \cup \beta : Y}$$

we can prove

$$\begin{aligned} \text{red: } (\neg a) & \qquad \text{if red to blue is allowed} \\ \text{blue: } (\neg c) & \qquad \text{if blue to red is allowed} \\ \text{blue: } (\neg b) & \qquad \text{if blue to blue is allowed.} \end{aligned}$$

The flattening function has to flatten

$$\begin{gathered} \{\text{red: } b, \text{ blue: } (\neg b)\} \\ \{\text{blue: } a, \text{ red: } (\neg a)\} \\ \{\text{blue: } c, \text{ (blue: } (\neg c) \text{ is not allowed!)}\} \end{gathered}$$

We first consider the case where red is stronger than blue that is, *blue to red* not allowed: we get b and $(\neg a)$ and c. Now we consider the case where blue is stronger than red, that is, *red to blue* not allowed. We get

$$\begin{gathered} \{\text{blue: } a, \text{ (red: } (\neg a) \text{ not allowed)}\} \\ \{\text{blue: } c, \text{ blue: } (\neg c)\} \\ \{\text{red: } b, \text{ blue: } (\neg b) \text{ if } c \text{ is available}\}. \end{gathered}$$

We cannot decide between c and $(\neg c)$ since both are blue. If we leave them both out or take $(\neg c)$ then $(\neg b)$ will not be obtainable and hence we will have $\{a, b\}$.

Observe that in this labeled formulation we have more options than when considering the valued-based argumentation frameworks mentioned above. On one hand, we can have $X, \neg X$ or neither as choices. On the other hand the label color (value) can itself be a whole database and so arguments about the values and their strengths can also be part of the system. Valued-based argumentation frameworks are only one level.

Moral debate example

We now consider an example referred to in [19] and attributed to Coleman [57] and Christie [55].

> “Hal, a diabetic, loses his insulin in an accident through no fault of his own. Before collapsing into a coma, he rushes to the house of Carla, another diabetic. She is not at home but Hal enters her house and uses some of her insulin. Was Hal justified, and does Carla have a right to compensation?”

The following are the arguments involved in the situation described above:

A. Hal is justified, since a person has a privilege to use the property of others to save their life - the case of necessity.

B. It is wrong to infringe the property rights of another.

C. Hal compensates Carla.

D. 1. If Hal is too poor to compensate Carla, he should nonetheless be allowed to take the insulin, as no one should die because they are poor.
2. Moreover, since Hal would not pay compensation if too poor, neither should he be obliged to do so even if he can.

E. Poverty is no defense for theft.

F. Hal is endangering Carla’s life.

G. Carla has abundant insulin.

H. Carla does not have ample insulin.

Argument D was originally stated as just one argument, but it has been splitted herein for later discussion. This example is depicted in Figure 11.5. Note that $\mathrm{H} = \neg \mathrm{G}$.

The following value properties are given to the arguments:

Life:	A, D, F
Property:	B, C, E
Fact:	G, H

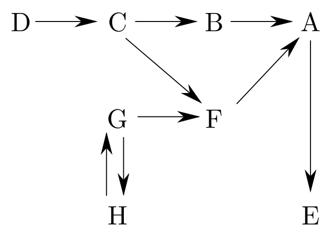


Figure 11.5: Moral debate example

It is arguable whether life is stronger than property or not but facts are always the strongest. Since H = ¬G, and since the facts cannot both hold simultaneously, that part of Figure 11.5 is regarded as a case of uncertainty.

The model needed for a proper analysis of this kind of problem in general (though maybe not necessarily the Hal problem) is a time/action model. There is a difference of values depending at what stage of the action sequence we are at. Has Hal entered Carla's house? Has he checked for insulin? Is it all over and Carla is dead? Each of these cases may have a different argument diagram, possibly with values depending on the previous one. The need for time/action models has already been strongly emphasized in [109] in connection with puzzles involved in the logical analysis of conditionals. This is factors of connected to contrary-to-duty models (see [158]), and also needed to incorporate uncertainty.

We require a better metalevel hierarchy of values and rules, as are available in labeled deductive systems. Such options can possibly also be made available to the abstract argumentation model via self-fibring.

The links $(X \rightarrow Y)$ should be given strength labels to help the modeling of more realistic cases where an argument X is attacked by arguments $Y_1, \ldots, Y_k$ with strength measuring $m_1, \ldots, m_k$. The link $X \rightarrow Y$ can be read as preventive action of X to stop Y. Thus, by giving probability of success, any acyclic network turns into a Bayesian one. This will introduce uncertainty into the framework. Actually, the probability of success is inversely proportional to the conditional probability of Y on X.

Bayesian aspects of the moral debate example

Recall Figure 11.5. We require a time/action model and contrary-to-duty considerations, as follows. We imagine an agent, such as Hal, who has available a stock of optional actions. These actions have the form

$$\mathbf{a} = (A, (B^+, B^-))$$

where A is the precondition of the action and B^+, B^- are the post-conditions. A must hold in order for Hal to be allowed to perform the action, in which case the resulting state is guaranteed to satisfy B^+. However, the agent may take the action anyway, without permission (that is, A does not hold), in which case the post-condition is B^-. Note that in most cases $B^- = B^+$.

Assume a state (or time) T_0, described by a logical theory Δ, where the actions that can be performed are

$$\mathbf{a}_1, \mathbf{a}_2, \dots \mathbf{a}_i = (A_i, (B_i^+, B_i^-)), \dots.$$

If $\Delta \vdash A_i$, then action $\mathbf{a}_i$ is allowable at time (state) T_0, otherwise not. If we perform the action $\mathbf{a}$, with post-condition B (B is either B^+ or B^-) then we move to time T_1, with state $\Delta_{\mathbf{a}} = \Delta \circ B$ where $\Delta \circ B$ is the revision of Δ by B. We have $\Delta \circ B \vdash B$.

Thus, to have time action model we need the following: a language for the theories Δ to describe states, a language for pre-condition and a language for post-conditions for actions, a logic or algorithm for determining when $\Delta \vdash A$ holds, where A is a pre-condition, and a revision algorithm giving for each Δ and post-condition B a new theory $\Delta' = \Delta \circ B$. Note that the languages for Δ, the pre-conditions and the post-conditions need not be the same.

The flow of time is future branching and it is generated by the actions. So, if, for example, an agent can perform actions $\mathbf{a}_1, \dots, \mathbf{a}_k$ as options, then after two steps in which he performs, say, $\mathbf{a}_1$ first and then $\mathbf{a}_3$, we may get a situation as the one depicted in Figure 11.6.

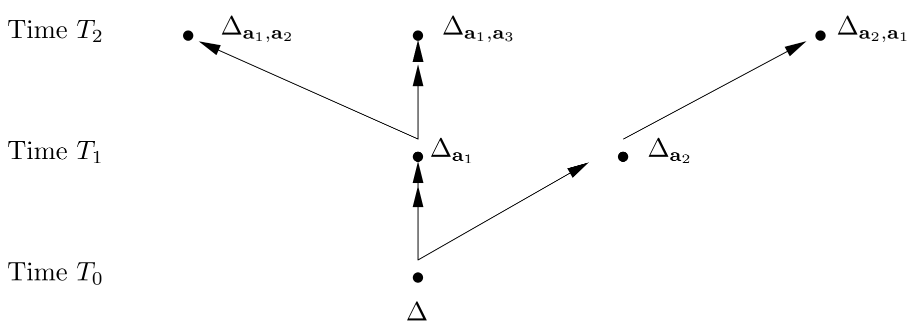


Figure 11.6: Flow of time generated by actions

The real history at time T_2 is

$$(\Delta, \Delta_{\mathbf{a}_1}, \Delta_{\mathbf{a}_1,\mathbf{a}_3}).$$

The states $\Delta_{\mathbf{a}_1,\mathbf{a}_2}$ and $(\Delta_{\mathbf{a}_2}, \Delta_{\mathbf{a}_2,\mathbf{a}_1})$ are hypotheticals.

At time T_0, the agent chose to take action $\mathbf{a}_1$ moving onto state $\Delta_{\mathbf{a}_1}$, but he could have chosen to take action $\mathbf{a}_2$ and done action $\mathbf{a}_1$ afterward, ending up at state $\Delta_{\mathbf{a}_2,\mathbf{a}_1}$ at time T_2. In reality, however, he chose to perform $\mathbf{a}_1$ and then $\mathbf{a}_3$.

The pre-conditions of actions can talk about states and hypotheticals. They need not be in the same language as Δ or the same language as the post-conditions. What is important are the algorithms for "$\vdash$" and "$\circ$".

We now analyze the moral debate example. We propose some probabilities as an example and we conclude by translating the statements A–H presented above into our time/action set up.

We first recast the situation in a more realistic way. Hal needs insulin. So does Carla. Both are poor and get their insulin from the Health Service. They get it in batches, though not at the same time. So the question whether Carla has spare insulin (G) depends on the time, and is a matter of probability. Hall loses all his insulin and would need to break into Carla's property to get hers. He has the option of calling the Health Service and asking for replacement, which he can use either for himself if it arrives immediately or to replace Carla's if necessary. He might get some money from friends. One thing is clear to him. If he steals Carla's insulin, it will complicate matters; it might be more difficult to find a replacement. So the question of compensation C is also a matter of probability.

The following are the possible scenarios. If property is valued more than life, then if Hal steals Carla's insulin, the probability of getting a replacement is lower in the case where Carla's life is not threatened. If life is valued more than property, his chances of obtaining replacement is higher in case Carla's life is threatened.

We must clarify what "getting a replacement" means. Hal will probably start a process for getting insulin for himself immediately at start time T_0. Since it might not arrive in time, he will break into Carla's home and use hers, and hope to use the insulin he "ordered" to replace Carla's. If Carla has ample insulin, there is a higher chance or that the replacement will arrive in time before Carla's life is threatened. If Carla does not have ample insulin, Hal can use this as a further reason to rush the process of replacement. This further reason might be counterproductive if property is valued above life.

Thus, the statement C, that is,

Hal gets a replacement

should be taken as

Hal gets a replacement before Carla is in need of it.

We may then have the following scenarios where P stands for probability $P(x)$ and it should be indexed by case and time, that is, $P_{1,a}, P_{1,b}, P_{2,a}$ and $P_{2,b}$:

Case 1. Property stronger than life

(a) Time = Before Hal breaks into Carla's house.

$$P(G) = \frac{2}{3} \qquad P(\neg G) = \frac{1}{3} \qquad P(C/G) = 0.9 \qquad P(\neg C/G) = 0.1$$
$$P(C/\neg G) = 0.5 \qquad P(\neg C/\neg G) = 0.5$$

Since Carla has ample insulin, Hal has more time to replace what he might take. Admittedly, Carla's life is in danger but there may not be enough time to get a replacement. On the other hand, this fact might help get the insulin more quickly. Event C means "getting replacement in time".

(b) Time = After Hal breaks into Carla's house.

At this stage the value of G is known: either $G = 1$ or $G = 0$. We get

$$P(C/G = 1) = 0.7 \qquad P(\neg C/G = 1) = 0.3$$
$$P(C/G = 0) = 0,4 \qquad P(\neg C/ G = 0) = 0.6$$

Less than before breaking into the house, because Hal committed a serious crime. He may not be favorable with the authority.

Case 2. Property not stronger than life

(a) Time = Before Hal breaks into Carla's house
Similar to Case 1, but $P(C/\neg G) = 0.9$ and $P(\neg C/\neg G) = 0.1$.

(b) Time = After Hal breaks into Carla's house
Similar to (a), but $P(C/G = 0) = 0.7$ and $P(\neg C/G = 0) = 0.3$.

We now translate the arguments involved in the original moral debate example. When is Hal justified in breaking into Carla's home? The answer is "yes" only in the case that life is stronger than property and he can reasonably say he is not risking her life. That depends on finding a replacement. We therefore have to calculate the probability of C given all the data we have. Thus, our time/action axis has the form depicted in Figure 11.7:

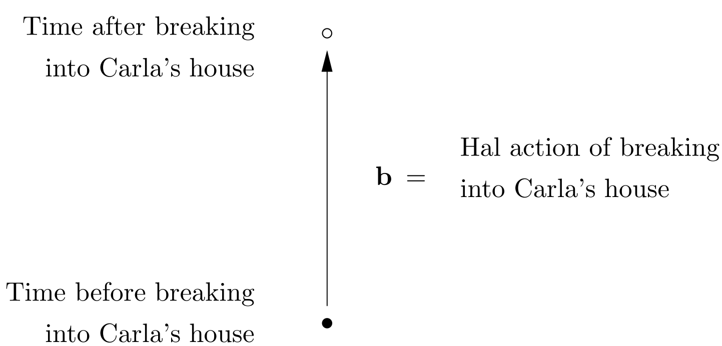


Figure 11.7: Time/action axis

The actions available to Hal are the following:

- **b** = breaking into Carla's house.

  The post-condition is breaking in and taking the insulin. The pre-condition of **b** is high probability of replacing Carla's insulin (in time before she needs it) in case *life is stronger than property* and $\perp$ (falsity, that is, no permission to do the action) in case *life is not stronger than property*.

- $\mathbf{r}$ = actions having to do with getting a replacement of insulin.

  We assume he can perform these actions at any time but the post-conditions are not clear.

Observe that we may need a temporal language for the post-conditions so that we can say something like "insulin will be delivered in two days". We need also agree the value of the threshold probability, for instance, only if there is at least 0.9 chance of replacement can Hal break into Carla's home to take the insulin.

Consider now the argument B referred to above: "It is wrong to infringe the property of others". It is an argument reflected in the pre-condition of the action $\mathbf{b}$. The action can be done when B satisfied otherwise not. We would write it as

$$\mathbf{b} = (\text{Justification, Break in and taking insulin}).$$

We now model the chain of events as a Bayesian network. Depending on the probability $P(G)$, Hal decides whether he wants to break into Carla's house $\mathbf{b}$ (no use breaking into her house if she does not have enough insulin). He is justified J in breaking $\mathbf{b}$ into Carla's house if there is high probability of compensation C. Thus, C depends both on $\mathbf{b}$ and G, and $\mathbf{b}$ also depends on G. The network depicted in Figure 11.8 describes the situation.

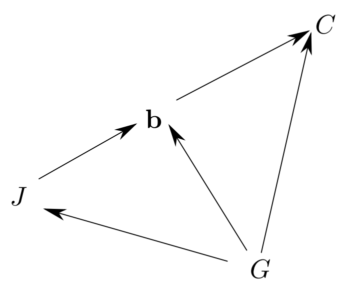


Figure 11.8: Modeling the chain of events as a Bayesian network

There are two problems with this representation.

- The dependency of $\mathbf{b}$ on G is not on $G = 1$ or $G = 0$ but on $P(G)$. Say if $P(G) < 0.1$ then maybe $\mathbf{b} = 0$. This is OK because the probabilities can be made to take account of that. This is allowed in the theory of Bayesian nets.

- The probabilities in Figure 11.8 depend on whether property is stronger than life or not. The best way to represent this is to have a Bayesian network with one variable only: *Case*.

  Case = 1 means *property is stronger than life* and *Case* = 0 means *property is not stronger than life*. For each case we get a different copy of Figure 11.8 with different probabilities. So, we get a substitution of the network of

Figure 11.8 into a one point network: *Case*. This operation is in accordance with the ideas in [276].

We can also allow for several justification variables to make it more realistic.

It is not difficult to work out the details of the other arguments $C, \ldots, H$.

Neural representation of argumentation frameworks

We now outline how to represent in neural nets any value-based argumentation framework involving x and anti-x (that is, arguments and counter-arguments). It capitalizes in the work presented in [71].

One possibility of representing the intended value-based argumentation framework in neural networks is to use neural-symbolic learning systems [78]. Neural networks are able to efficiently represent (and learn) multi-part, cumulative argumentation, as exemplified below.

Cumulative behavior can be encoded in neural-symbolic learning systems with the use of a hidden layer of neurons in addition to an input and an output layer in a feedforward network. Rules of the form

$$(A \wedge B) \rightarrow C$$

can be represented by connecting input neurons that represent concepts A and B to a hidden neuron, say h_1, and then connecting h_1 to an output neuron that represents C in such a way that output neuron C is activated (true) if input neurons A and B are both activated (true). If, in addition, a rule

$$B \rightarrow C$$

is also to be represented, another hidden neuron h_2 can be added to the network to connect input neuron B to output neuron C in such a way that C is now activated also if B alone is activated. This is illustrated in Figure 11.9. The network can be used to perform the computation of the rules in parallel such that C is true whenever B is true.

In a neural network, positive weights can represent the support for an argument, while negative weights can be seen as an attack on an argument.

Hence, a negative weight from a neuron A to a neuron B can be used to implement the fact that A attacks B. Similarly, a positive weight from B to itself can be used to indicate that B supports itself. Since we concentrate on feedforward networks, neuron B will appear on both the input and the output layers of this network as shown in Figure 11.10, in which dotted lines are used to indicate negative weights. In this neural network A attacks B via h_1, while B supports itself via h_2.

Suppose now that, in addition, B attacks C. We need to connect input neuron B to output neuron C via a new hidden neuron h_3. Since B appears on both the network's input and output, we also need to add a feedback connection from

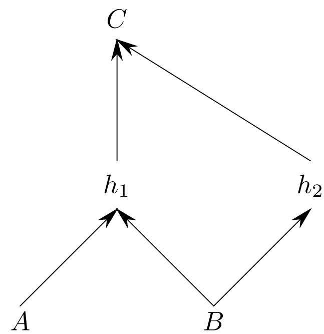


Figure 11.9: Using hidden neurons

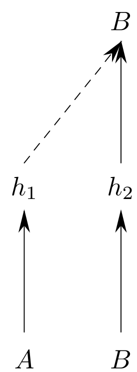


Figure 11.10: Using negative weights for counter-argumentation

output neuron B to input neuron B such that the activation of B can be computed by the network according to the chain "A attacks B", "B attacks C", etc.

As a result, in Figure 11.11 if the attack from A on B is stronger (according to the network's weights) than B's support to itself, then A will block the activation of (output) B, and (input) B will not be able to block the activation of C. Note that, for simplicity, we do not represent B's feedback connection. The network's final computation will include C and not B in a stable state. If, on the other hand, A is not strong enough to block B, then B will be activated and block C.

Let us take the example in which an argument A attacks an argument B, and B attacks an argument C, which in turn attacks A in a cycle. In order to implement this in a neural network, we need positive weights to explicitly represent the fact that A supports itself, B supports itself and so does C. In addition, we need negative weights from A to B, from B to C and from C to A (see Figure 11.12) to implement attacks.

If all the weights are the same in absolute terms, no argument wins, as one would expect, and the network stabilizes with neither A, nor B, nor C activated.

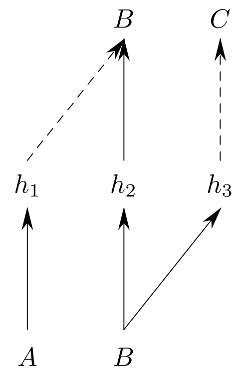


Figure 11.11: Computation of arguments and counter-arguments

If, however, the value of A (that is, the weight from h_1 to A) is stronger than the value of C (the weight from h_3 to C, which is expected to be the same in absolute terms as the weight from h_3 to A), C cannot attack and defeat A. As a result, A is activated. Since A and B have the same value (as, for instance, in the previous case of an unspecified priority), B is not activated, since the weights from h_1 and h_2 to B will both have the same absolute value. Finally, if B is not activated then C will be activated, and a stable state $\{A, C\}$ will be reached in the network. Note that the order in which we reason does not affect the final result (the stable state reached). For example, if we started from B successfully attacking C, C would not be able to attack A, but then A would successfully attack B, which would this time round not be able to successfully attack C, which in turn would be activated in the final stable state $\{A, C\}$. This indicates that a neural (parallel) implementation of this reasoning process could be advantageous, also from a purely computational point of view.

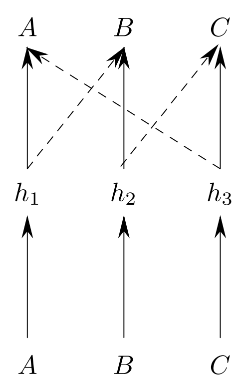


Figure 11.12: The moral-debate example as a neural network

Note that, as in the general case of argumentation networks, also in the case of neural networks we can extend Bench-Capon's model with the use of self-fibring neural networks, which allow for the recursive substitution of neural networks inside nodes of other networks [79].

The implementation of the network's behavior (weights and biases) must be such that, when we start from a number of positive arguments (input vector $\{1, 1, \ldots, 1\}$), weights with the same absolute values cancel each other producing zero as the output neuron's input potential. A neuron with zero or less input potential is then deactivated, while a neuron with positive input potential is activated. This allows for the implementation of the argumentation framework in neural-symbolic learning systems, in the style of the translation algorithms developed at [80].

Self-fibring of argumentation networks

We now concentrate in self-fibring of argumentation networks. We begin with the network in Figure 11.4. We pick a node in it, say node a, and substitute another network for that node. Assume we choose the network depicted in Figure 11.5. We thus get the "network" in Figure 11.13.

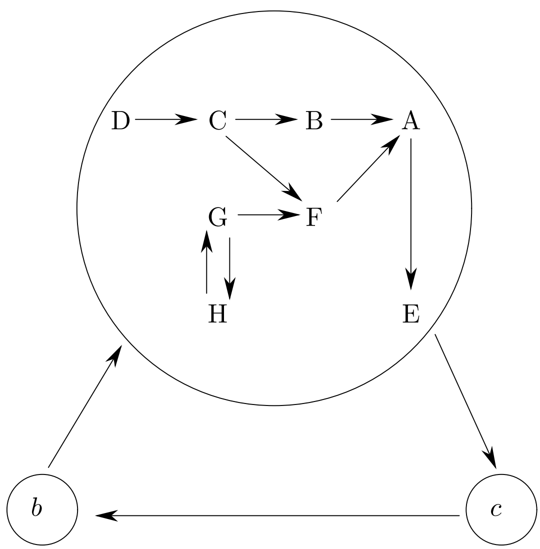


Figure 11.13: Self-fibring argumentation networks

The need of self-fibring may arise if additional arguments, supporting the contents of the node, are available. The self-fibring problem has three important aspects that we discuss in the sequel: intuitive meaning, formal aspect and coherence.

What is the intended interpretation/meaning of this substitution? This can be decided by the needs of the application area. There are several options. One option is to consider that node a is an argument, so Figure 11.5 can be viewed as delivering some winning argument (A of Figure 11.5) which can combine/support a. Another option is to consider that Figure 11.5 represents a network so b of Figure 11.4 can plug into it. We can connect b to all (or some) members in Figure 11.4 and similarly connect all (or some) members in Figure 11.5 into c of Figure 11.4. For other possibilities see [276, 79, 108].

We can consider syntactical and semantic substitutions. With respect to syntactical substitution, formally, the node a is supposed to be an argument. So we need a fibring function

$$\mathbf{F}(node, network) = e$$

yielding a node e and so we end up with Figure 11.14. $\mathbf{F}$ might do, for example,

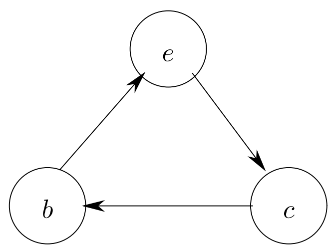


Figure 11.14: Syntactical substitution

the following: $\mathbf{F}$ can use the colour of node a to modify the colours of the nodes in Figure 11.5 (the substituted network), and maybe also modify some connections in Figure 11.5, and then somehow emerge with some winning argument e and a colour to be substituted/combined with a and its colour. With respect to semantic substitution, if the original network has an interpretation, then the node a can get several possible semantic values. We can make the definition of the substitution context sensitive to those values. We may even go to the extent of substituting different networks for different options of values.

Finally, we refer to coherence. To enable successful repeated recursive substitution of networks within networks, we have to modify our definition of the original network. One possibility consists in extending the notion of network and allow arrows to either support or defeat arguments. Another possibility is to restrict the substitution of networks for nodes by compatibility/consistency conditions. Assume we have a set of nodes and links of the form (a, b) meaning that a attacks b. Assume we also have valuation colors. A weaker color cannot attack a stronger color. Let a be a node. Define the notion of x is a supportive (resp. attacking) node for a as follows:

- a is supportive of a;
- if x is supportive (resp. attacking) node of a and y attacks x then y is an attacking (resp. supportive) node of a.

Now let a be a node in a network A and suppose we have another network N which we want to substitute for a. We must assume a appears in N with the same color value as it is in A. We substitute N for a and make new connection as follows:

- any node x of A which attacks a in A is now connected to any node y in N which supports a in N;
- any node y in N which supports a in N is now made connected to any node x of A which a of A is attacking.

This definition is reasonable. Node a is an argument in network A and N is another network which is supposed to support a (a in N). Thus, anything which attacks a in A will attack of all a supporters in N and these in turn will attack whatever nodes a attacks in A. Note that he may be attacking facts in N by this wholesale connection of arrows.

11.4 Software specification

In this section, we point out that the fibring techniques can also be used in the context of software specification. As a first example we can refer to the simple case of dynamic logic (see [142]). Dynamic logic has been used for reasoning about programs, namely showing that a particular specification is met by a particular program.

In the sequel, we briefly describe some combination mechanisms that have been proposed in the context of software specification. Both temporalization and parameterization are particular cases of fibring that nevertheless are worthwhile to explore for themselves. Synchronization is a different mechanism endowed with a deductive and a semantic component. Results were obtained for proving soundness and completeness. Institutions were introduced in [129] as a mean to describe in an abstract way the semantic component of a logic with the aim of defining operations on specifications. Later on π-institutions were introduced in the same spirit but for the deductive component (see [95]). Combinations of institutions were studied namely by introducing the category of institutions. The problem with combination of institutions is that mixing of connectives from different logics is not allowed. Parchments [130, 212] were then considered, again only with a semantic component. Fibring is worthwhile to be imported to the context of parchments namely for adding a deductive component and studying preservation of metaproperties.

11.4.1 Temporalization and parameterization

Temporalization (see [96, 97]) and parameterization (see [42]) are less richer combination mechanisms than fusion and fibring. Namely, the underlying operator is neither commutative nor associative or even idempotent. The general basic idea is that, given logics $\mathcal{L}'$ and $\mathcal{L}''$ we define a new logic $\mathcal{L}$ where each signature C

of $\mathcal{L}$ is such that $C_0 = L(C') \cup C''_0$ and $C_k = C''_k$ for every $k \geq 1$. In the case of temporalization, $\mathcal{L}''$ is temporal logic and $\mathcal{L}'$ is the logic we want to temporalize. In the case of parameterization, $\mathcal{L}''$ can be any logic.

Soundness and completeness preservation were analyzed for temporalization. Below we briefly describe parameterization.

Parameterization generalizes the temporalization construction. Roughly speaking, it consists of replacing an atomic part of one logic $\mathcal{L}$ by another logic $\mathcal{L}'$. Logic $\mathcal{L}$ is the parameterized logic, the atomic part is the formal parameter and logic $\mathcal{L}'$ is the actual parameter logic. Parameterization is a particular case of constrained fibring.

As an example of application, we can refer to the case were we intend to describe a state based system as, for instance, a data base. Data base dynamics is easily described using a propositional based modal logic $\mathcal{L}_{pml}$ but data base states are better described using first-order logic $\mathcal{L}_{fol}$. The goal is to use a suitable combination of $\mathcal{L}_{pml}$ and $\mathcal{L}_{fol}$ to describe the data base. When combining both logics using parameterization, we get a new logic whose formulas are obtained by replacing propositional constants in formulas of $\mathcal{L}_{pml}$ by first-order formulas. In this logic we can freely apply modalities but we cannot apply quantifiers to modal formulas. Hence, we do not get the full first-order modal logic, as it would be the case if we considered fibring as in Chapter 6. This asymmetry is the essential distinction between parameterization and fibring. The semantic structures of the new logic are Kripke structures where the usual valuation for propositional constants is replaced by a "zooming in" map [23] associating a first-order semantic structure with a fixed assignment to each state. From the deductive point of view, instantiation of axioms and rules of $\mathcal{L}_{pml}$ with pure first-order formulas is allowed but first-order reasoning cannot be applied to formulas with modalities.

We now give a more detailed account of parameterization from the deductive point of view. The following capitalizes in [42]. In the sequel, we assume signatures, Hilbert calculi and related notions as introduced in Chapter 2.

To begin with, we introduce some preliminary notions. Let $f : C' \to C''$ be a signature morphism. We define the set of *f-monoliths* as the set of ground formulas

$$gL(C'')_f = \{(c(\varphi_1, \ldots, \varphi_k)) \in gL(C'') : c \notin f(C'_k), k \in \mathbb{N}\}.$$

Given a set R of inference rules over C'', we denote by $R\!\uparrow_f$ the set of all rules $\langle \Delta, \varphi \rangle$ in R such that φ and every $\delta \in \Delta$ only involve connectives in $f(C')$.

We are now ready to define the notion of parameterization. Consider Hilbert calculi

- $H' = \langle C', R' \rangle$;
- $H'' = \langle C'', R'' \rangle$;

and

- a signature $\overline{C}$ such that $\overline{C} \leq C'$;

- a set P such that $P \subseteq C'_0$ and $P \cap \overline{C}_0 = \emptyset$;
- a pair $h = \langle h_1, h_2 \rangle$ where
  - $h_1 : \overline{C} \to C''$ is a signature morphism;
  - $h_2 : P \to gL(C'')_{h_1}$ is a surjective map.

We say that H' is the *parameterized calculus*, $\overline{C}$ is the *shared signature* and P is the *formal parameter*. The Hilbert calculus H'' is the *actual parameter*. The morphism h_1 is the *shared connectives matching* and the map h_2 is the *parameter passing map*.

The parameterized signature, C', and the actual parameter signature, C'', may share connectives. The signature $\overline{C}$ identifies the connectives in C' we want to share, and h_1 sets the matching with the corresponding connectives in C''. Observe that h_1-monoliths are precisely the ground formulas of the actual parameter whose main connective is not a shared one. For simplicity, we assume in the sequel that $h_1(\overline{C}_0) \cap gL(C'')_{h_1} = \emptyset$. On the other hand, the constant symbols in P identify the atomic part of $L(C')$ that we allow to be replaced by formulas of the actual parameter. These formulas are the h_1-monoliths. The parameter passing map says which h_1-monolith replaces each constant in P.

The *parameterization of H' with H'' according to h* is the Hilbert calculus

$$H = \langle C, R \rangle$$

where

- $C_0 = (C'_0 \backslash (\overline{C}_0 \cup P)) \cup h_1(\overline{C}_0) \cup gL(C'')_{h_1}$,
  $C_k = (C'_k \backslash \overline{C}_k) \cup h_1(\overline{C}_k)$ for $k > 0$;
- $R = \{\langle \Delta^h, \varphi^h \rangle : \langle \Delta, \varphi \rangle \in R'\} \cup R'' \!\uparrow_{h_1}$
  $\cup \{\langle \rho(\Delta), \rho(\varphi) \rangle : \langle \Delta, \varphi \rangle \in R'' \backslash R'' \!\uparrow_{h_1} \text{ and } \rho : \Xi \to gL(C'')\}$;

where, for each $\varphi \in L(C')$, φ^h is the formula we obtain from φ by replacing all the occurrences of symbols c in $\overline{C}$ or P by $h_1(c)$ or $h_2(c)$, respectively.

Observe that in the resulting calculus $H = (C, R)$ the formal parameter symbols in P are dropped and h_1-monoliths become constant symbols of signature C. The other non shared connectives of the parameterized signature C' are kept in C. The shared connectives are replaced by the corresponding matching ones. Hence, to obtain the formulas in the mixed language $L(C)$, the only C'' connectives that can be freely used are the ones that are shared with C' connectives. The other C'' connectives can only occur in h_1-monoliths which are formulas in $L(C'')$ and, in $L(C)$, are just constant symbols. This is quite different from the fibring situations where any connective of both components (possibly with a different name) can be freely used to obtain the formulas of the fibring.

With respect to the inference rules, the calculus H keeps all the rules of the parameterized calculus H', but, as expected, with shared C' connectives replaced by the corresponding matching ones, and the formal parameter constants in P replaced by the corresponding h_1-monoliths. The calculus H also includes all the rules in the actual parameter H'' that only involve shared connectives. With respect to the H'' rules that involve some non shared connective, only its ground instances (with respect to $L(C'')$) are included in the resulting calculus. Only these instances are included because, otherwise, derivations in H could involve unwanted instantiations as, for example, instantiations involving formulas $(c(\varphi_1, \ldots, \varphi_n))$ where c is one of those non shared connectives in C'' and some φ_i is a ground formula in $L(C')$. As already referred to above, these kind of formulas are not allowed in the mixed language $L(C)$.

We now illustrate the parameterization construction with the example presented at the beginning: we parameterize a Hilbert calculus $H_{\mathbf{K}}$, corresponding to propositional modal logic **K**, with a Hilbert calculus H_{FOL}, corresponding to classic first-order logic. As expected, only the connectives $\Rightarrow$ and $\neg$ are shared. Thus, quantifiers can not be freely used in the resulting language. They can only occur in monoliths. Moreover, in the resulting calculus, we will only have ground instances of rules involving quantifiers. Hence, for our purposes herein, we do not need the all setting for first-order based logics presented in Chapter 6, namely, rules with provisos. We can just use signatures and Hilbert calculi as defined in Chapter 2.

In the sequel we consider fixed a set X of variables such that $X \cap \Xi = \emptyset$.

Example 11.4.1 Let $H_{\mathbf{K}} = \langle C^{\mathbf{K}}, R_{\mathbf{K}} \rangle$ be the following Hilbert calculus, corresponding to propositional modal logic **K**:

- $C_0^{\mathbf{K}} = \mathbb{P}$, $C_1^{\mathbf{K}} = \{\neg, \Box\}$, $C_2^{\mathbf{K}} = \{\Rightarrow\}$ and $C_k^{\mathbf{K}} = \emptyset$ for $k > 2$;
- $R_{\mathbf{K}}$ consists of the following rules
  - $\langle \emptyset, (\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$ Ax1
  - $\langle \emptyset, ((\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_3)) \Rightarrow ((\xi_1 \Rightarrow \xi_2) \Rightarrow (\xi_1 \Rightarrow \xi_3))) \rangle$ Ax2
  - $\langle \emptyset, (((\neg\, \xi_1) \Rightarrow (\neg\, \xi_2)) \Rightarrow (\xi_2 \Rightarrow \xi_1)) \rangle$ Ax3
  - $\langle \emptyset, ((\Box(\xi_1 \Rightarrow \xi_2)) \Rightarrow ((\Box\xi_1) \Rightarrow (\Box\xi_2))) \rangle$ K
  - $\langle \{\xi_1, (\xi_1 \Rightarrow \xi_2)\}, \xi_2 \rangle$ MP
  - $\langle \{\xi_1\}, (\Box\xi_1) \rangle$ Nec

Let $H_{\mathrm{FOL}} = \langle C^{\mathrm{FOL}}, R_{\mathrm{FOL}} \rangle$ be the following Hilbert calculus, corresponding to classical first-order logic, where $F = \{F_k\}_{k \in \mathbb{N}}$ and $Q = \{Q_k\}_{k \in \mathbb{N}}$ are respectively the family of sets of function symbols and the family of set of predicate symbols:

- $C_0^{\mathrm{FOL}} = \{q(t_1, \ldots, t_k) : q \in Q_k \text{ and } t_1, \ldots, t_k \in T\}$,
  $C_1^{\mathrm{FOL}} = \{\neg\} \cup \{\forall x : x \in X\}$, $C_2^{\mathrm{FOL}} = \{\Rightarrow\}$ and $C_k^{\mathrm{FOL}} = \emptyset$ for $k > 2$;
  where T is inductively defined as follows:
  (i) $X \cup F_0 \subseteq T$; (ii) $\{f(t_1, \ldots, t_k) : f \in F_k \text{ and } t_1, \ldots, t_k \in T\} \subseteq T$;

- R_{FOL} consists of the following rules
  - Ax1, Ax2, Ax3, MP
  - $\langle \emptyset, ((\forall x \varphi) \Rightarrow \varphi_t^x) \rangle$ Ax4
    for all $t \in T$ free for $x \in X$ and $\varphi \in gL(C^{\text{FOL}})$
  - $\langle \emptyset, ((\forall x(\varphi \Rightarrow \xi_1)) \Rightarrow (\varphi \Rightarrow (\forall x \xi_1))) \rangle$ Ax5
    for all $x \in X$ not free in $\varphi \in gL(C^{\text{FOL}})$
  - $\langle \{\xi_1\}, (\forall x \xi_1) \rangle$, for all $x \in X$ Gen

Let

- $\overline{C}$ be the shared signature
  where
  $\overline{C}_0 = \emptyset$, $\overline{C}_1 = \{\neg\}$, $\overline{C}_2 = \{\Rightarrow\}$ and $\overline{C}_k = \emptyset$ for $k > 2$;
- $\mathbb{P}$ be formal parameter;
- $h = \langle h_1, h_2 \rangle$ such that
  - $h_1 : \overline{C} \to C^{\text{FOL}}$ is an inclusion;
  - $h_2 : \mathbb{P} \to gL(C^{\text{FOL}})_{h_1}$ is a surjective map.

Note that the set of h_1-monoliths is

$$gL(C^{\text{FOL}})_{h_1} = \{q(t_1, \ldots, t_k) : q \in Q_k \text{ and } t_1, \ldots, t_k \in T\} \\ \cup \{(\forall x \varphi) : (\forall x \varphi) \in gL(C^{\text{FOL}})\}.$$

The parameterization of $H_{\mathbf{K}}$ with H_{FOL} according to h is the Hilbert calculus $H = \langle C, R \rangle$ where

- $C_0 = gL(C^{\text{FOL}})_{h_1}$, $C_1 = \{\neg, \Box\}$, $C_2 = \{\Rightarrow\}$ and $C_k = \emptyset$ for $k > 2$;
- R consists of the following rules
  - Ax1, Ax2, Ax3, K, MP, Nec
  - $\langle \emptyset, ((\forall x \varphi) \Rightarrow \varphi_t^x) \rangle$
    for all $t \in T$ free for $x \in X$ and $\varphi \in gL(C^{\text{FOL}})$
  - $\langle \emptyset, ((\forall x(\varphi \Rightarrow \psi)) \Rightarrow (\varphi \Rightarrow (\forall x \psi))) \rangle$
    for all $\varphi, \psi \in gL(C^{\text{FOL}})$ and $x \in X$ not free in φ
  - $\langle \{\varphi\}, (\forall x \varphi) \rangle$
    for all $\varphi \in gL(C^{\text{FOL}})$ and $x \in X$. ∇

In [42], a categorial characterization of parameterization is presented, as well as some transfer results for soundness, completeness and decidability.

11.4.2 Synchronization

Herein, we give a brief description of synchronization on models (see [238]). In order to understand the mechanism, we start by recalling the concept of satisfaction system.

A satisfaction system is a triple $\langle L, M, \Vdash\rangle$ where L is a set (of formulas), M is a class (of models) and $\Vdash \subseteq M \times L$ is a relation (the satisfaction relation). Given two satisfaction systems S' and S'', the basic idea of synchronization on models is to obtain a new satisfaction system S where each model is a pair composed by a model of S' and a model of S'', but, may be not all of them. In a sense, synchronization is a general form of product extended to logics not necessarily modal and where not all models in the Cartesian product $M' \times M''$ are present.

In the sequel, we give a more detailed account of synchronization on models. Consider the satisfaction systems $S' = \langle L', M', \Vdash'\rangle$ and $S'' = \langle L'', M'', \Vdash''\rangle$, and let $R \subseteq M' \times M''$ be a relation (the synchronization relation). The *synchronization on models* of S' and S'' by R is the satisfaction system

$$S = \langle L' \cup L'', R, \Vdash\rangle$$

where $\Vdash \subseteq R \times (L' \cup L'')$ is such that

$$\langle m', m''\rangle \Vdash \varphi' \text{ if } m' \Vdash' \varphi' \quad \text{and} \quad \langle m', m''\rangle \Vdash \varphi'' \text{ if } m'' \Vdash'' \varphi''$$

for $m' \in M'$, $m'' \in M''$, $\varphi' \in L'$ and $\varphi'' \in L''$.

As an example, assume that S' is a satisfaction system for propositional linear temporal logic and S'' is a satisfaction system for unsorted equational logic.

A model m' for S' is a map $m' : \mathbb{N} \to \mathbb{P}$. The notion of satisfaction of a formula φ' at natural number n by a model m', denoted by

$$m', n \Vdash' \varphi'$$

is defined as follows:

- $m', n \Vdash' p$ if $p \in m'(n)$ whenever $p \in \mathbb{P}$;
- $m', n \Vdash' (\neg \psi')$ if $m', n \not\Vdash' \psi'$;
- $m', n \Vdash' (\psi'_1 \Rightarrow \psi'_2)$ if either $m', n \not\Vdash' \psi'_1$ or $m', n \Vdash' \psi'_2$;
- $m', n \Vdash' (X\psi')$ if $m', n+1 \Vdash' \psi'$;
- $m', n \Vdash' (G\psi')$ if $m', k \Vdash' \psi'$ for every $k \geq n$.

Then we can define $\Vdash'$ as follows:

$$m' \Vdash' \varphi \text{ if } m', n \Vdash' \varphi \text{ for every } n \in \mathbb{N}.$$

With respect to S'', let us consider a signature $\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$ where each Σ_n is a set (of operators of arity n). Then, as usual, the set of terms $T_\Sigma(X)$ is the least set generated by Σ and a fixed set X (of variables). The set L is composed by all equations $(t_1 \approx t_2)$ where t_1, t_2 are terms. A model m'' for S'' is an algebra over Σ, that is, a pair $A = \langle U, \mu\rangle$ where U is a set and $\nu = \{\nu_n\}_{n\in\mathbb{N}}$ such that each $\nu_n : \Sigma_n \to U^{U^n}$ is a map. Given an assignment $\rho : X \to U$, the denotation of a term t, $[\![t]\!]^\rho_A$, is an element of U and it is defined as usual. Finally, the satisfaction relation $\Vdash''$ is as follows:

$$m'' \Vdash'' (t_1 \approx t_2) \;\text{ if }\; [\![t_1]\!]^\rho_A = [\![t_2]\!]^\rho_A \text{ for every assignment } \rho.$$

For the purpose of defining the synchronization relation, we are going to assume that $\{a_u : u \in T_\Sigma(\emptyset)\} \subseteq \mathbb{P}$ where $T_\Sigma(\emptyset)$ is the set of all terms that do not involve variables in X. The relation R is defined as follows

$$\langle m', m''\rangle \in R \text{ if and only if } m' \Vdash' (a_{u_1} \Leftrightarrow a_{u_2}) \text{ whenever } m'' \Vdash'' (u_1 \approx u_2).$$

It is worthwhile to see how the semantic entailment reflects the intended synchronization. We verify that

$$\{a_{g(c)}, (g(x) \approx h(x))\} \vDash a_{h(c)}.$$

Assume that $\langle m', m''\rangle$ is such that $\langle m', m''\rangle \Vdash a_{g(c)}$ and $\langle m', m''\rangle \Vdash (g(x) \approx h(x))$. Then,

- $m'' \Vdash'' (g(x) \approx h(x))$ by hypothesis;
- $m'' \Vdash'' (g(c) \approx h(c))$ by soundness of substitution in S'';
- $m' \Vdash' (a_{g(c)} \Leftrightarrow a_{h(c)})$ by definition of R;
- $m' \Vdash' a_{g(c)}$ by hypothesis;
- $m' \Vdash' a_{h(c)}$ by reasoning in S';
- $\langle m', m''\rangle \Vdash a_{h(c)}$ by definition of $\Vdash$.

In [238], a form of deductive synchronization based on formulas is also investigated. In [239], sufficient conditions for the preservation of soundness and completeness are proved.

11.4.3 Specifications on institutions

Several ways of relating and combining logical systems in the framework of institutions [129, 131] have been proposed, having in mind heterogeneous specifications, that is, specifications written in different institutions. Among them we can refer to institution morphisms [131], maps [207], representations [255] and simulations [13]. Institution morphisms, for example, describe how an institution can be built over a simpler one, whereas representations show how an institution can be encoded in a more elaborate one. The combination mechanism provided by institution morphisms provides combination of sets of formulas but does not allow the combination of individual formulas, that is, formulas involving connectives from different logics are not allowed. To overcome this problem, parchments [130] and parchment morphisms [210] were considered. Roughly speaking, parchments are algebraic presentations of institutions where the formulas are presented as terms over an algebraic signature, models are sets of algebraic signatures morphisms and satisfaction relies on the initiality of the term algebra. Some variants have also been proposed, namely λ-parchments [211], model-theoretic parchments [212], and **c**-parchments [39, 38]. The notion of **c**-parchment was put forward as a way to bring the fibring mechanism, as well as results for preservation of relevant properties, to the framework of institutions. This kind of parchments is an extension of model-theoretic parchments where the algebras of truth values are endowed with a closure operator.

Herein, we refer to fibring of **c**-parchments by describing the fibring of **c**-rooms. In fact, a **c**-parchment can be seen as a particular family of **c**-rooms indexed by (abstract) signatures. Technically, this means that a **c**-parchment can be also defined as a functor from the category of such signatures to the category of **c**-rooms.

In the sequel, we consider algebraic many-sorted signatures $\Sigma_\phi = \langle S_\phi, O\rangle$ where S_ϕ is the set of sorts with a distinguished sort ϕ, and $O = \{O_u\}_{u\in S_\phi^+}$ is a family of sets of operators indexed by their type. The carrier set of ϕ in the free algebra $T(\Sigma_\phi)$ over Σ_ϕ (the term algebra) corresponds to the set of formulas. Given an algebra $\mathcal{A}$ over Σ_ϕ, the denotation of a term t in $\mathcal{A}$, $[\![t]\!]^{\mathcal{A}}$, is defined as expected.

We denote by $\mathrm{cAlg}(\Sigma_\phi)$ the class of all pairs $\langle \mathcal{A}, \mathbf{c}\rangle$ with $\mathcal{A}$ an algebra over Σ_ϕ and $\mathbf{c}$ a closure operation on $\mathcal{A}_\phi$, the carrier set of sort ϕ. Recall from Section 1.1 of Chapter 1 that $\mathbf{c} : \wp(\mathcal{A}_\phi) \to \wp(\mathcal{A}_\phi)$ is extensive, monotonic and idempotent. The set $\mathcal{A}_\phi$ constitutes the set of truth values.

Given a morphism $h : \Sigma'_\phi \to \Sigma''_\phi$ and an algebra $\mathcal{A}$ over Σ''_ϕ, $\mathcal{A}|_h$ is the reduct of $\mathcal{A}$ with respect to h, that is, the algebra over Σ'_ϕ where the carrier set for each sort s is $\mathcal{A}_{h(s)}$, and the interpretations of the operators are as their images by h in $\mathcal{A}$.

We now introduce the notion of **c**-room. A **c**-*room* is a pair

$$R = \langle \Sigma_\phi, M\rangle$$

where $M \subseteq \mathrm{cAlg}(\Sigma_\phi)$. A **c**-room induces the following local entailment relation:

$$\Phi \vDash^\ell \psi \ \text{ if } [\![\psi]\!]^{\mathcal{A}} \in \{[\![\varphi]\!]^{\mathcal{A}} : \varphi \in \Phi\}^{\mathbf{c}} \text{ for every } \langle \mathcal{A}, \mathbf{c}\rangle \in M$$

where $\Phi \cup \{\psi\} \subseteq T(\Sigma_\phi)_\phi$. With respect to global entailment relation the definition is as follows:

$$\Phi \vDash^g \psi \ \text{ if } \ [\![\psi]\!]^{\mathcal{A}} \in \emptyset^{\mathbf{c}} \text{ whenever } \{[\![\varphi]\!]^{\mathcal{A}} : \varphi \in \Phi\} \subseteq \emptyset^{\mathbf{c}} \text{ for every } \langle \mathcal{A}, \mathbf{c}\rangle \in M.$$

We now turn to the fibring of **c**-rooms. Consider **c**-rooms $R_1 = \langle\langle S_{\phi\,1}, O_1\rangle, M_1\rangle$ and $R_2 = \langle\langle S_{\phi\,2}, O_2\rangle, M_2\rangle$. We assume that the shared signature is

$$\langle S_{\phi\,0}, O_0\rangle$$

with $S_{\phi\,0} = S_{\phi\,1} \cap S_{\phi\,2}$, and $O_{0,u} = O_{1,u} \cap O_{2,u}$ for $u \in S^+_{\phi\,0}$ is shared according to the corresponding signature inclusion morphisms. Let

$$R_0 = \langle\langle S_{\phi\,0}, O_0\rangle, M_0\rangle$$

where $M_0 = \text{cAlg}(\langle S_{\phi\,0}, O_0\rangle)$. The (constrained) *fibring of* R_1 *and* R_2 is the **c**-room

$$R_1 \circledast R_2 = \langle\langle S_\phi, O\rangle, M\rangle$$

such that:

- $S_\phi = S_{\phi\,1} \cup S_{\phi\,2}$;
- $O_u = \begin{cases} O_{1,u} \cup O_{2,u} & \text{if } u \in S^+_{\phi\,0} \\ O_{i,u} & \text{if } u \in S^+_{\phi\,i} \setminus S^+_{\phi\,0} \text{ for } i \in \{1,2\} \\ \emptyset & \text{otherwise;} \end{cases}$
- $M \subseteq \text{cAlg}(\langle S_\phi, O\rangle)$ is the class of all pairs $\langle \mathcal{A}, \mathbf{c}\rangle$ such that, for $i \in \{1,2\}$, $\langle \mathcal{A}|_{h_i}, \mathbf{c}\rangle \in M_i$, where $h_i = \langle f_i, g_i\rangle$ is the signature morphism whose components are the inclusions $f_i : S_{\phi\,i} \to S_\phi$ and $g_i : O_i \to O$.

In the fibring $R_1 \circledast R_2$, the set M consists of all those pairs $\langle \mathcal{A}, \mathbf{c}\rangle$ that can be obtained by joining together any two pairs $\langle \mathcal{A}_1, \mathbf{c}_1\rangle \in M_1$ and $\langle \mathcal{A}_2, \mathbf{c}_2\rangle \in M_2$ such that $\mathcal{A}_{1\,s} = \mathcal{A}_{2\,s} = \mathcal{A}_s$ for every $s \in S_{\phi\,0}$, $o_{\mathcal{A}_1} = o_{\mathcal{A}_2} = o_{\mathcal{A}}$ for every $o \in O_{0,u}$ with $u \in S^+_{\phi\,0}$, and $\mathbf{c}_1 = \mathbf{c}_2 = \mathbf{c}$. The fibring of **c**-rooms R_1 and R_2 corresponds to a pushout in the category of **c**-rooms.

For propositional based logics, we can also define deduction rooms

$$\langle\langle\{\phi\}, O\rangle, R_g, R_\ell\rangle$$

where R_g and R_ℓ are sets of respectively global and local inference rules, similar to the Hilbert calculi introduced in Chapter 2. Then, given two deduction rooms $\langle\langle\{\phi\}, O_1\rangle, R_{g_1}, R_{\ell 1}\rangle$ and $\langle\langle\{\phi\}, O_2\rangle, R_{g_2}, R_{\ell 2}\rangle$ their fibring is

$$\langle\langle\{\phi\}, O_1\rangle, R_{g_1}, R_{\ell 1}\rangle \circledast \langle\langle\{\phi\}, O_2\rangle, R_{g_2}, R_{\ell 2}\rangle = \langle\langle\{\phi\}, O\rangle, R_g, R_\ell\rangle$$

where O is defined as above, $R_g = R_{g_1} \cup R_{g_2}$ and $R_\ell = R_{\ell 1} \cup R_{\ell 2}$. Derivations and all the related notions are defined as expected.

The notions of soundness and completeness of a deductive room for a **c**-room are also defined as expected. In [39], preservation results for soundness and completeness by fibring, in this propositional setting, are presented. Those results are extended beyond the propositional case in [38].

When dealing with heterogeneous specifications, the notion of heteromorphism between algebras is very useful. Given a signature morphism $\sigma : \Sigma \to \Sigma'$, and algebras $\mathcal{A}$ and $\mathcal{A}'$ over Σ and Σ', respectively, we say that $h : \mathcal{A} \to \mathcal{A}'$ is an *heteromorphism* if $h : A \to A'$ is a map between the carriers of the algebras that is a homomorphism between $\mathcal{A}$ and the reduct $\mathcal{A}'|_\sigma$. Based on this notion, a solution to the collapsing problem for global reasoning was introduced in [40].

11.5 Emergent applications

In this section, we briefly point out new applications where we can investigate the advantages of fibring techniques. Among them we refer to security, quantum computation and information and space-time. In all these applications there is a mixture of operators that have a different flavor. Moreover, it seems that new operators have to be introduced and studied.

Security

Among the first logical accounts of security, we can refer to the authentication logic (now known as BAN logic) introduced in [32]. In this logic, we have formulas like the following:

- $P| \equiv X$ for "Principal P believes statement X";
- $P \triangleleft X$ for "Principal P sees X";
- $P |\!\sim X$ for "Principal P once said statement X";
- $\stackrel{K}{\mapsto} P$ for "Principal P has K as public key";
- $P \stackrel{K}{\leftrightarrow} Q$ for "Principals P and Q may use the shared key K";
- $\{X\}_K$ for "Statement X encrypted under the key K".

An example of a rule is the following:

$$\frac{P| \equiv P \stackrel{K}{\leftrightarrow} Q \qquad P \triangleleft \{X\}_K}{P| \equiv Q |\!\sim X}$$

meaning that if P believes that key K is shared with Q and sees a message X encrypted under K, then P believes that Q once said statement X. Hence, we can detect different aspects that can be modeled by belief and temporal operators among others. Hence, a possible methodology would be to start by isolating the different components and choosing a particular logic for each component. Then, study the metaproperties. Afterward define the fibring of the component logics and analyze the preservation of the metaproperties. Finally, include the interaction axioms and testing if the original logic is recovered.

More recently, a logic was proposed in [141] for reasoning about information hiding in general and anonymity in particular, in the context of multiagent systems.

Assume that we have n agents $i = 1, \ldots, n$ and e is the environment agent. Consider the signature C of our logic to be defined as follows:

- $C_0 = \mathbb{P}$;
- $C_1 = \{\neg\} \cup \{\Box_i : i = 1, \ldots, n\}$;
- $C_2 = \{\Rightarrow\}$.

Let $L(C)$ to be defined as follows: $C_0 \subseteq L(C)$, $(\neg\,\varphi), (\Box\varphi) \in L(C)$ whenever $\varphi \in L(C)$, $(\varphi_1 \Rightarrow \varphi_2) \in L(C)$ whenever $\varphi_1, \varphi_2 \in L(C)$. In $\mathbb{P}$ we have propositional symbols of the form θ_i^a where $i = 1, \ldots, n$. The elements of the set

$$\{a : \theta_i^a \in \mathbb{P}\}$$

are called *actions*.

The propositional symbol θ_i^a stands for "agent i has performed action a or will perform a in the future".

We provide a brief account of the semantics. We assume that an interpretation structure is a pair $\langle S, r\rangle$ where S is a set (of states) and $r : \mathbb{N} \to S^{n+1}$ is a map (a run). A run r associates a global state $\langle s_e, s_1, \ldots, s_n\rangle$ to each $n \in \mathbb{N}$ and we denote by $r_i(n)$ the state s_i corresponding to the (local) state of agent i at the point n of run r. An interpretation system is a pair

$$I = \langle \mathcal{R}, \pi\rangle$$

where $\mathcal{R}$ is a set of interpretation structures and $\pi : \mathcal{R} \times \mathbb{N} \to \wp\mathbb{P}$ is a map. The satisfaction of $\varphi \in L(C)$ by I at $\langle S, r\rangle \in \mathcal{R}$ and n, denoted by

$$I, \langle S, r\rangle, n \Vdash \varphi$$

is then inductively defined as follows:

- $I, \langle S, r\rangle, n \Vdash \varphi$ if $\varphi \in \pi(\langle S, r\rangle, n)$ for $\varphi \in \mathbb{P}$;
- $I, \langle S, r\rangle, n \Vdash (\neg\psi)$ if $I, \langle S, r\rangle, n \nVdash \psi$;
- $I, \langle S, r\rangle, n \Vdash (\psi_1 \Rightarrow \psi_2)$ if either $I, \langle S, r\rangle, n \nVdash \psi_1$ or $I, \langle S, r\rangle, n \Vdash \psi_2$;
- $I, \langle S, r\rangle, n \Vdash (\Box_i\varphi)$ if
  $I, \langle S', r'\rangle, n' \Vdash \varphi$ for all $\langle S', r'\rangle \in \mathcal{R}$, $n' \in \mathbb{N}$ such that $r(n) = r'(n')$.

An interpretation structure I satisfies φ if $I, \langle S, r\rangle, n \Vdash \varphi$ for all $\langle S, r\rangle \in \mathcal{R}$ and $n \in \mathbb{N}$.

With semantics we are able to express several kinds of anonymity. For instance, an action a performed by agent i is minimally anonymous with respect to agent j in an interpretation system I if

$$I \Vdash (\neg(\Box_j \theta_i^a)).$$

That is, agent j does not know if agent i performed or will perform action a. Hence, given a protocol described by a set of formulas, we can prove (semantically) whether or not it satisfies the anonymity assertions.

It seems worthwhile to study the properties of the underlying logic and we believe that fibring can help in this. But before it also seems interesting to investigate the properties of each $\Box_i$ as well as promoting θ_i^a to operators.

Also in [141], probabilities are used to express anonymity namely for quantifying the uncertainty of observers about the system. It is clear now that probabilities should be present in most security contexts. Another example is zero-knowledge protocols as we will know illustrate following the example in [202].

A zero-knowledge protocol is a protocol that allows a prover to show to a verifier that he has a secret without revealing it (for more details on zero-knowledge protocols see [136]). The protocol consists of three steps:

- First, the prover sends a value (commitment) to the verifier such that, if he has the secret, for any challenge put to him by the verifier, he is able to send a response convincing the verifier that he has the secret. If he is cheating, then, he cannot produce a response at least with probability $\frac{1}{2}$;
- Second, the verifier sends a random bit (challenge) to the prover;
- Finally, the prover sends a response according to the bit received.

In principle there are three objectives:

1. The verifier has probability 1 of verifying the secret, if indeed the prover has it (soundness);
2. The verifier has probability less than 1 of verifying the secret, if the prover is cheating (completeness);
3. The verifier can not learn the secret (security).

For describing the protocol above we can use classical propositional logic endowed with probability operator $\int$ and global connective $\sqsupset$ for relating assertions involving probability reasoning as well as propositional formulas. For instance, $(\int \varphi)$ where φ is a propositional formula means the probability of φ. From a semantic point of view we have to work with probability structures that are pairs $\langle V, \mu \rangle$ where V is a set of valuations and μ is a probability measure over $\wp V$. A probability structure $\langle V, \mu \rangle$ satisfies $(\int \varphi)$ if $\mu(\{v \in V : v \Vdash \varphi\}) = 1$. On the other hand, a probability structure $\langle V, \mu \rangle$ satisfies $(\psi_1 \sqsupset \psi_2)$ if either $\langle V, \mu \rangle \nVdash \psi_1$ or $\langle V, \mu \rangle \Vdash \psi_2$.

Finally, $\langle V, \mu \rangle \not\Vdash \varphi$, where φ is a classical propositional formula if $v \not\Vdash \varphi$ for every $v \in V$. For more details, including a Hilbert calculus, see [202].

Returning to the zero-knowledge protocol, let Π be $\{s, a, c\}$ where s is the propositional symbol for stating that the prover has a secret, a is the propositional symbol for stating that the verifier accepts the secret of the prover and finally, c is the propositional symbol for stating that the commitment is compatible with the challenge. The specification S of the zero-knowledge protocol is as follows:

$$\begin{array}{ll} S_1 & ((s \wedge c) \Rightarrow a) \\ S_2 & ((s \wedge (\neg\, c)) \Rightarrow a) \\ S_3 & (((\neg\, s) \wedge c) \Rightarrow (\neg\, a)) \\ S_4 & (((\neg\, s) \wedge (\neg\, c)) \Rightarrow a) \\ S_5 & ((\textstyle\int (\neg\, c)) = \frac{1}{2}) \end{array}$$

From S, we prove

$$\begin{array}{ll} O_1 & (s \sqsupset ((\textstyle\int a) = 1)) \\ O_2 & ((\neg\, s) \sqsupset ((\textstyle\int a) < 1)) \end{array}$$

corresponding to objectives 1 and 2 respectively.

We conclude by noting that it seems that an essential mechanism should be provided for combining logics with probability.

Also for security applications, some new operators are also relevant. Namely, we illustrate the almost everywhere quantifier as introduced and analyzed in [67]. Assume that we have a first-order signature $\langle F, P \rangle$. To the usual set of formulas we add the formula $(\mathsf{Æ} x \varphi)$ to be read as "almost everywhere φ". Semantically speaking we have to enrich first-order structures with a measurable component.

An interpretation structure is a tuple

$$\mathfrak{M} = \langle D, [\![\cdot]\!], \mathcal{B}, \mu \rangle$$

where:

- D is a non-empty set;
- $\langle D, [\![\cdot]\!] \rangle$ is a first-order interpretation structure, that is:
  - for each $f \in F_n$, $[\![f]\!] : D^n \to D$;
  - for each $p \in P_n$, $[\![p]\!] : D^n \to \{0, 1\}$.
- $\langle D, \mathcal{B}, \mu \rangle$ is a measure space, that is:
  - $\mathcal{B}$ is a σ-algebra over D;
  - μ is a measure on $\mathcal{B}$.
- $\mu(D) \neq 0$.

Satisfaction in a structure $\langle D, [\![\cdot]\!], \mathcal{B}, \mu\rangle$ given a variable assignment ρ is defined in the usual way as for FOL, with the following extra clause:

$$\mathfrak{M}, \rho \Vdash \text{Æ}x\varphi \text{ if there is } B \in \mathcal{B} \text{ such that } (D \setminus |\varphi|^x_{\mathfrak{M}\rho}) \subseteq B \text{ and } \mu(B) = 0$$

where $|\varphi|^x_{\mathfrak{M}\rho}$ (the extent of φ relative to x in model $\mathfrak{M}$ with assignment ρ) is defined by

$$|\varphi|^x_{\mathfrak{M}\rho} = \{d \in D : \mathfrak{M}, \rho^x_d \Vdash \varphi\}$$

where ρ^x_d is the assignment such that $\rho^x_d(y) = \rho(y)$ for $y \neq x$ and $\rho^x_d(x) = d$.

It seems worthwhile to fiber such an operator with, for instance, knowledge operators.

Quantum computation

A quantum logic was first proposed in [22]. Recently, there is a growing interest in quantum logic motivated by quantum information and computation. Among relevant contributions, we can refer to [14] and [201]. Herein we give the flavor of quantum logic as presented in [201]. For this purpose, we introduce possible axioms for the Schrödinger cat problem.

The relevant attributes of the cat are: being inside or outside the box, alive or dead, and moving or still. We use the quantum bits *cat-in-box*, *cat-alive* and *cat-moving* for representing these three attributes. The formulas constrain the state of the cat at different levels of detail:

1. $[\mathit{cat\text{-}in\text{-}box}, \mathit{cat\text{-}alive}, \mathit{cat\text{-}moving}]$;
2. $(\mathit{cat\text{-}moving} \Rightarrow \mathit{cat\text{-}alive})$;
3. $((\Diamond\, \mathit{cat\text{-}alive}) \sqcap (\Diamond\, (\neg\, \mathit{cat\text{-}alive})))$;
4. $(\boxminus[\mathit{cat\text{-}alive}])$;
5. $(\int \mathit{cat\text{-}alive} = \frac{1}{3})$;
6. $([\mathit{cat\text{-}alive}, \mathit{cat\text{-}moving}]\Diamond\ (\mathit{cat\text{-}alive} \wedge \mathit{cat\text{-}moving}) : \frac{1}{\sqrt{6}},$
   $(\mathit{cat\text{-}alive} \wedge (\neg\, \mathit{cat\text{-}moving})) : \frac{1}{\sqrt{6}},$
   $((\neg\, \mathit{cat\text{-}alive}) \wedge (\neg\, \mathit{cat\text{-}moving})) : e^{i\frac{\pi}{3}}\sqrt{\frac{2}{3}})$.

We provide an intuitive interpretation of the formulas above. Assertion 1 states that the quantum bits *cat-in-box*, *cat-alive* and *cat-moving* are not entangled with the other quantum bits of the cat system. Assertion 2 is a classical constraint on the set of admissible valuations: if the cat is moving then it is alive. Assertion 3 states the famous paradox: the cat can be in a state where it is possible that the cat is alive and it is possible that the cat is dead. Assertion 4 states that the quantum bit *cat-alive* is entangled with other quantum bits. Assertion 5 states that the cat is in a state where the probability of observing it alive (after collapsing the wave function) is $\frac{1}{3}$. Finally, assertion 6 states that the quantum bits *cat-alive* and *cat-moving* are not entangled with other quantum bits and that in the quantum state there is a classical valuation with amplitude $\frac{1}{\sqrt{6}}$ where the cat is alive and moving, there is another classical valuation also with amplitude $\frac{1}{\sqrt{6}}$ where the cat is alive and not moving, and there is a classical valuation with amplitude $e^{i\frac{\pi}{3}}\sqrt{\frac{2}{3}}$ where the cat is dead (and, thus, thanks to 2, also not moving).

For better understanding of the formulas above, recall that the states of an isolated quantum bit are vectors of the form $z_0|0\rangle + z_1|1\rangle$ in a Hilbert space where $z_0, z_1 \in \mathbb{C}$ and $|z_0|^2 + |z_1|^2 = 1$. In other words, they are unit vectors in the (unique up to isomorphism) Hilbert space of dimension two. In the logic presented in [201], each quantum bit is represented by a propositional symbol (called a qubit symbol). Furthermore, each qubit state (called quantum valuation) should be a superposition of the two possible classical valuations.

Given a quantum valuation $|\psi\rangle$ and a classical valuation v, the inner product $\langle v|\psi\rangle$ is said to be the logical amplitude of $|\psi\rangle$ for v. Logical amplitudes are important to evaluate probabilities. For example, if the system is in the particular state

$$\alpha_{00\omega_1}|00\omega_1\rangle + \alpha_{01\omega_2}|01\omega_2\rangle + \alpha_{01\omega_3}|01\omega_3\rangle + \alpha_{10\omega_4}|10\omega_4\rangle$$

then the probability of observing the first two qubits $\mathsf{qb}_0, \mathsf{qb}_1$ in the classical valuation 01 is given by $|\alpha_{01\omega_2}|^2 + |\alpha_{01\omega_3}|^2$.

The logic in [201] is a classical propositional logic endowed with probability, with global operators like $\sqsupset$ and terms for denoting logical amplitudes. It is shown there to be weakly complete.

We observe that a challenge would be to combine logics with probability (once more) and quantum mechanics features.

Space-time

Logics that put together space and time are nowadays of increasing interest. There are two main streams. One of them is dedicated to adding topological operators to temporal logic (see [172, 2]). These logics are generally called dynamic topological logics and were introduced in [171, 170, 173] and in [12].

In order to illustrate a logic endowed with such operators we refer to [119, 165, 166]. In these papers, temporal logic is endowed with some metric operators. For instance, we have the following formulas $(\exists^{\leq a}\varphi)$ and $(\forall^{\leq a}\varphi)$ where a is a positive rational number. Semantically speaking, the appropriate structure is a metric model, that is, a pair $\langle\langle W, d\rangle, V\rangle$ where $\langle W, d\rangle$ is metric space, and $V : \mathbb{P} \to \wp W$ is a map. We say that $\langle\langle W, d\rangle, V\rangle$ satisfies formula ψ at point w, denoted by $\langle\langle W, d\rangle, V\rangle, w \Vdash \psi$, if:

- $\langle\langle W, d\rangle, V\rangle, w \Vdash (\exists^{\leq a}\varphi)$ if there exists $w' \in W$ such that $d(w, w') \leq a$ and $\langle\langle W, d\rangle, V\rangle, w' \Vdash \varphi$;

- $\langle\langle W, d\rangle, V\rangle, w \Vdash (\forall^{\leq a}\varphi)$ if $\langle\langle W, d\rangle, V\rangle, w' \Vdash \varphi$ for every $w' \in W$ such that $d(w, w') \leq a$.

From a semantic point of view, we can interpret the operators above as well as the linear temporal operators $\bigcirc, \Box, \Diamond$ in a structure

$$\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle$$

where:

- $\langle\langle W, d\rangle, V\rangle$ is a metric model;
- $f : W \to W$ is an isometric map;
- $\mu \leq \omega$ is an ordinal.

We say that a structure $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle$ satisfies δ in $w \in W$ after $m \leq \mu$ iterations where m is a finite ordinal, denoted by

$$\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash \delta$$

whenever

- $\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash p$ if $w \in V(p)$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\neg\,\varphi)$ if $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \not\Vdash \varphi$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\varphi_1 \Rightarrow \varphi_2)$
  if $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash \varphi_2$ or $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\neg\,\varphi_1)$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\exists^{\leq a}\varphi)$ if $\langle\langle W, d\rangle, V\rangle, w \Vdash (\exists^{\leq a}\varphi)$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\forall^{\leq a}\varphi)$ if $\langle\langle W, d\rangle, V\rangle, w \Vdash (\forall^{\leq a}\varphi)$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\bigcirc\varphi)$ if $\langle\langle W, d\rangle, V\rangle, f(w), m+1 \Vdash \varphi$ and $m < \mu$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\Box\varphi)$
  if $\langle\langle W, d\rangle, V\rangle, f^n(w), m+n \Vdash \varphi$ for every $n \geq 0$ such that $m+n \leq \mu$;
- $\langle\langle\langle W, d\rangle, V\rangle, f, \mu\rangle, w, m \Vdash (\Diamond\varphi)$
  if $\langle\langle W, d\rangle, V\rangle, f^n(w), m+n \Vdash \varphi$ for some $n \geq 0$ such that $m+n \leq \mu$.

It seems worthwhile to investigate the properties of the metric operators per se and use the fibring techniques in order to be able to prove metaproperties about the whole logic. That is, one can study on one hand the logic for time and, on the other hand, the logic for space and then use combining tools to obtain the space-time logic.

The interested reader can also have a look at [246, 5] for applications of space-time to physics.

11.6 Outlook

We start by referring to topics in fibring that need further investigation. Then, we will point out new forms of combination that should be developed in order to cope with emergent applications. In most cases, the combination is not fully logical in the sense that it is not just pure combinations of logics.

Perhaps the most challenging one is related to capture logics whose deductive components do not induce a consequence operator, in the sense of Section 1.1 of Chapter 1. Once this problem is solved we will be able to combine logics like linear logics and we shall also know how to combine different kinds of sequent calculi (where the antecedent and the consequent of a sequent can be a list, a multiset, a set, etc) endowed with different kinds of rules. As we referred to in Chapter 4, we believe that using polycategories, as introduced in [254], can be a possible solution (generalizing the approach in [175]). It may be the case that some fundamental research in polycategories has still to be done.

Once the problem above is better understood, it seems worthwhile to develop the semantic counterpart and to analyze metaproperties in this more general setting.

In what concerns state-of-the-art fibring, we believe that there are some issues worthwhile to be pursued, namely, preservation of metaproperties such as weak completeness, finite model, decidability and complexity. In what concerns weak completeness we believe that the techniques used in the Chapter 4 will be of use to this problem. In what concerns the finite model property and decidability it seems worthwhile to try to generalize the known and useful techniques that are available for modal logics. Another issue is related with the relaxation of the metatheorem of deduction aiming at providing sufficient conditions for the preservation of completeness of more logics.

Moreover, it seems worthwhile to investigate fibring of sequent calculus labeled with truth-values and obtain preservation results for cut elimination.

As referred to in Chapter 6, we believe that the semantic counterpart for analyzing first-order based logics considered therein is not general enough, contrarily to the degree of abstraction that we use for both propositional based logics and higher-order logics. Generalizing cylindrical algebras to the more general setting and discussing therein the generalized Barcan formulas seem to be an investigation direction worthwhile to pursue.

As we pointed out before, the techniques of fibring seem to be ready to cope with many applications. However, fibring has to be complemented in order to be useful to some emergent applications. In all of them the problem can be described as follows: we start with a logic $\mathcal{L}$ (that can already be the fibring of other logics) and we want to enrich the language with a new (non-logical) operator. This operator can be a probability operator, a quantum operator or a space operator. In the case of probability and quantum operators there is no mixture between the connectives of the logic and either probabilities or quantum aspects. That is, for instance, we can apply probabilities to the formulas in the logic $\mathcal{L}$ but we cannot apply connectives in $\mathcal{L}$ to formulas involving the probability operator. We call $p\mathcal{L}$ the logic resulting from the enrichment of $\mathcal{L}$ with a probability operator. In the case of space operators, in general we can fully mix them with the connectives of $\mathcal{L}$.

We now detail some issues related to the introduction of probability operators, that we call probabilization of a logic. Assume that we have a logic $\mathcal{L}$ with a language L, a Hilbert calculus H and a class of models M. The probabilization of a logic involves the following steps:

- We have three steps to enrich the language:
  - considering terms of the kind $(\int \varphi)$, referring to the probability of φ where $\varphi \in L$;
  - choosing the adequate subset of the set of real numbers or, alternatively, choosing a real-closed field so that we can have additional terms to be compared with probability terms;
  - adopting some classical logic to reason over comparisons of terms.
- The semantic structures for the probability logic $p\mathcal{L}$ are triples $\langle M', \mathcal{B}, \mu\rangle$ where $M' \subseteq M$, $\mathcal{B}$ is a σ-algebra over $\wp M'$ and $\mu : \mathcal{B} \to [0,1]$ is a probability measure.
- In what concerns denotation of terms, we say that
$$[\![(\textstyle\int \varphi)]\!] = \mu(\{m \in M' : m' \Vdash_{\mathcal{L}} \varphi\}).$$
- With respect to the Hilbert calculus of $p\mathcal{L}$, we have to introduce some axioms related to probabilities, namely: $\mu(\psi) = 1$ for every formula $\psi \in L$ such that $M' \Vdash \psi$.

We observe that the axioms for representing probability aspects depend on the logic $\mathcal{L}$. In particular, probabilities are well suited for propositional logic in the sense that it is possible to relate the probability of a formula with the probabilities of its subformulas. For instance the following equivalence holds:

$$((\int (\neg\,\varphi)) \Leftrightarrow (1 - (\int \varphi))).$$

However, things are not so easy when we consider modal logic and, in particular, the formula $(\int (\Box\varphi))$.

An interesting problem is related to the transference of metaproperties. For instance, which additional conditions should be imposed in order to get a complete $p\mathcal{L}$ when $\mathcal{L}$ is complete. In [202] a proof of preservation of completeness is presented when $\mathcal{L}$ is classical propositional logic. The logic for relating comparisons of terms considered therein is also a generalized classical propositional logic. The proof of preservation of completeness capitalizes in properties of classical logic such as having normal forms.

Bibliography

[1] T. Ahed. Algebraic logic, where does it stand today? *The Bulletin of Symbolic Logic*, 11(4):465–516, 2005.

[2] M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors. *Handbook of Spatial Logic*. Springer, 2007.

[3] H. Albert. *Traktat über kritische Vernunft*. Mohr Siebeck, Tübingen, 1968. English translation: Treatise on Critical Reason, Princeton University Press, Princeton,1985.

[4] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, 1985.

[5] H. Andréka, J. Madarász, and I. Németi. Logical axiomatizations of space-time. Samples from the literature. In *Non-Euclidean Geometries*, volume 581 of *Mathematical Applications*, pages 155–185. Springer, 2006.

[6] N. Angelopoulos and S. Muggleton. Machine learning metabolic pathway descriptions using a probabilistic relational representation. *Electronic Transactions in Artificial Intelligence*, 7(9), 2002.

[7] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *The Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[8] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic*, 124(1-3):287–299, 2003.

[9] A. Armando, editor. *Frontiers of Combining Systems, 4*, volume 2309 of *Lecture Notes in Computer Science*. Springer, 2002.

[10] S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb, and J. Woods, editors. *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 1. King's College Publications, 2005.

[11] S. Artemov, H. Barringer, A. d'Avila Garcez, L. Lamb, and J. Woods, editors. *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 2. King's College Publications, 2005.

[12] S. Artemov, J. Davoren, and A. Nerode. Modal logics and topological semantics for hybrid systems. Technical report, University of Cornell, 1997.

[13] E. Astesiano and M. Cerioli. Relationships between logical frameworks. In M. Bidoit and C. Choppy, editors, *Recent Trends in Data Type Specification*, volume 655 of *Lecture Notes in Computer Science*, pages 126–143. Springer, 1993.

[14] A. Baltag and S. Smets. The logic of quantum programs. In P. Selinger, editor, *Proceeding of the 2nd Workshop on Quantum Programming Languages*, pages 39–56. Turku Centre for Computer Science, 2004.

[15] M. Barr and C. Wells. *Category Theory for Computing Science.* Prentice Hall, 1990.

[16] D. Basin, S. Matthews, and L. Viganò. Labelled propositional modal logics: theory and practice. *Journal of Logic and Computation*, 7(6):685–717, 1997.

[17] B. Beckert and D. Gabbay. Fibring semantic tableaux. In *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 1397 of *Lecture Notes in Computer Science*, pages 77–92. Springer, 1998.

[18] J. Bell. *Toposes and Local Set Theories.* Oxford University Press, 1988.

[19] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13:429–448, 2003.

[20] J. Bicarregui, T. Dimitrakos, D. Gabbay, and T. Maibaum. Interpolation in practical formal development. *Logic Journal of the IGPL*, 9(2):231–243, 2001.

[21] G. Birkhoff. *Lattice Theory.* AMS Colloquium Publications, 1967.

[22] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936.

[23] P. Blackburn and M. de Rijke. Zooming in, zooming out. *Journal of Logic, Language and Information*, 6(1):5–31, 1997.

[24] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Tracts in Theoretical Computer Science.* Cambridge University Press, 2001.

[25] W. Blok. On the degree of incompleteness of modal logics. *Bulletin of the Section of Logic of the Polish Academy of Sciences*, 7(4):167–175, 1978.

[26] G. Boolos. *The Logic of Provability.* Cambridge University Press, 1993.

[27] L. Borkowski, editor. *J. Łukasiewicz Selected Works.* Studies in Logic and the Foundations of Mathematics. Elsevier, North Holland, 1970.

[28] B. Boutsinas and M. Vrahatis. Artificial nonmonotonic neural networks. *Artificial Intelligence*, 132:1–38, 2001.

[29] J. Bueno-Soler and W. Carnielli. Possible-translations algebraization for paraconsistent logics. *Bulletin of the Section of Logic, University of Lodz*, 34(2):77–92, 2005.

[30] J. Bueno-Soler, M. Coniglio, and W. Carnielli. Finite algebraizability via possible-translations semantics. In Carnielli et al. [50], pages 79–86.

[31] J. Bueno-Soler, M. Coniglio, and W. Carnielli. Possible-translations algebraizability. In J.-Y. Béziau, W. Carnielli, and D. Gabbay, editors, *Handbook of Paraconsistency,* Proceedings of the III World Congress on Paraconsistency, held in Toulouse 2003, volume 4 of *Studies in Logic and Practical Reasoning.* North Holland, 2007. To appear.

[32] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *Proceedings of the Royal Society London Series A*, 426(1871):233–271, 1989.

[33] S. Buss, editor. *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics.* Elsevier, North Holland, 1998.

[34] C. Caleiro. *Combining Logics.* PhD thesis, IST, Universidade Técnica de Lisboa, 2000.

[35] C. Caleiro, W. Carnielli, M. Coniglio, and J. Marcos. Two's company: "The humbug of many-logical values". In J.-Y. Béziau, editor, *Logica Universalis*, volume 1, pages 169–189. Birkhäuser, 2005.

[36] C. Caleiro, W. Carnielli, M. Coniglio, A. Sernadas, and C. Sernadas. Fibring non-truth-functional logics: Completeness preservation. *Journal of Logic, Language and Information*, 12(2):183–211, 2003.

[37] C. Caleiro, W. Carnielli, J. Rasga, and C. Sernadas. Fibring of logics as a universal construction. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 13, pages 123–187. Springer, 2nd edition, 2005.

[38] C. Caleiro, P. Gouveia, and J. Ramos. Completeness results for fibred parchments: Beyond the propositional base. In M. Wirsing, D. Pattinson, and R. Hennicker, editors, *Recent Trends in Algebraic Development Techniques*, volume 2755 of *Lecture Notes in Computer Science*, pages 185–200. Springer, 2003.

[39] C. Caleiro, P. Mateus, J. Ramos, and A. Sernadas. Combining logics: Parchments revisited. In M. Cerioli and G. Reggio, editors, *Recent Trends in Algebraic Development Techniques*, volume 2267 of *Lecture Notes in Computer Science*, pages 48–70. Springer, 2001.

[40] C. Caleiro and J. Ramos. Cryptofibring. In Carnielli et al. [50], pages 87–92.

[41] C. Caleiro, A. Sernadas, and C. Sernadas. Fibring logics: Past, present and future. In Artemov et al. [10], pages 363–388.

[42] C. Caleiro, C. Sernadas, and A. Sernadas. Parameterisation of logics. In J. Fiadeiro, editor, *Recent Trends in Algebraic Development Techniques*, volume 1589 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 1999.

[43] A. Carbone. The Craig interpolation theorem for schematic systems. In *Collegium Logicum*, volume 2, pages 87–100. Springer, 1996.

[44] A. Carbone. Interpolants, cut elimination and flow graphs for the propositional calculus. *Annals of Pure and Applied Logic*, 83(3):249–299, 1997.

[45] W. Carnielli. Many–valued logics and plausible reasoning. In *Proceedings of the XX International Congress on Many–Valued Logics, University of Charlotte, USA*, pages 328–335. IEEE Computer Society, 1990.

[46] W. Carnielli. Possible-translations semantics for paraconsistent logics. In D. Batens, C. Mortensen, G. Priest, and J. van Bendegem, editors, *Frontiers of Paraconsistent Logic: Proceedings of the I World Congress on Paraconsistency*, Logic and Computation Series, pages 149–163. Research Studies Press, King's College Publications, 2000.

[47] W. Carnielli and M. Coniglio. A categorial approach to the combination of logics. *Manuscrito*, 22(2):69–94, 1999.

[48] W. Carnielli and M. Coniglio. Splitting logics. In Artemov et al. [10], pages 389–414.

[49] W. Carnielli, M. Coniglio, and J. Marcos. Logics of formal inconsistency. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 14, pages 15–107. Kluwer, 2007. In print.

[50] W. Carnielli, M. Dionísio, and P. Mateus, editors. *Proceedings of CombLog'04, Workshop on Combination of Logics: Theory and Applications*, 1049-001 Lisboa, Portugal, 2004. DM, Instituto Superior Técnico.

[51] W. Carnielli and J. Marcos. A taxonomy of C-systems. In W. Carnielli, M. Coniglio, and I. D' Ottaviano, editors, *Proceedings of the 2nd World Congress on Paraconsistency 2000*, pages 1–94. Marcel Dekker, 2002.

[52] W. Carnielli and P. Mateus, editors. *Selected Papers from CombLog'04*, volume 13(6) – Special issue of *Logic Journal of the IGPL*, 2005.

[53] W. Carnielli, J. Rasga, and C. Sernadas. Preservation of interpolation features by fibring. *Journal of Logic and Computation.* In print.

[54] B. Chellas. *Modal Logic: An Introduction.* Cambridge University Press, 1980.

[55] G. Christie. *The Notion of an Ideal Audience in Legal Argument.* Kluwer, 2000.

[56] I. Cloete and J. M. Zurada, editors. *Knowledge-Based Neurocomputing.* The MIT Press, 2000.

[57] J. Coleman. *Risks and Wrongs.* Cambridge University Press, 1992.

[58] M. Coniglio. Categorical logic with partial elements. In W. Carnielli and I. D'Ottaviano, editors, *Advances in Contemporary Logic and Computer Science*, volume 235 of *Contemporary Mathematics*, pages 63–82. American Mathematical Society, 1999.

[59] M. Coniglio. Recovering a logic from its fragments by meta-fibring. *Logica Universalis*, 2(1):1–39, 2007.

[60] M. Coniglio and V. Fernández. Plain fibring and direct union of logics with matrix semantics. In B. Prasad, editor, *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, pages 1590–1608. IICAI, 2005.

[61] M. Coniglio, A. Martins, A. Sernadas, and C. Sernadas. Fibring (para)consistent logics. Research report, DM, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2000. Extended abstract. Presented at II World Congress on Paraconsistency - WCP'2000.

[62] M. Coniglio, A. Sernadas, and C. Sernadas. Fibring logics with topos semantics. *Journal of Logic and Computation*, 13(4):595–624, 2003.

[63] M. Coniglio and C. Sernadas. A complete axiomatization of higher-order intuitionistic logic. In F. Sautter and H. de Araújo-Feitosa, editors, *Lógica: Teoria, Aplicações e Reflexões*, volume 39 of *Coleção CLE*, pages 25–58. CLE - UniCamp, 2004.

[64] L. Couturat. *La Logique de Leibnitz.* Paris: Felix Alcan, 1901.

[65] W. Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *The Journal of Symbolic Logic*, 22:250–268, 1957.

[66] R. Crole. *Categories for Types.* Cambridge Mathematical Textbooks. Cambridge University Press, 1993.

[67] L. Cruz-Filipe, J. Rasga, A. Sernadas, and C. Sernadas. A complete axiomatization of discrete-measure almost-everywhere quantification. Technical report, CLC, DM, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2006.

[68] L. Cruz-Filipe, A. Sernadas, and C. Sernadas. Heterogeneous fibring of deductive systems via abstract proof systems. Technical report, CLC, DM, Instituto Superior Técnico, 2005.

[69] L. Cruz-Filipe and C. Sernadas. Sequent calculi based on derivations. Technical report, CLC, DM, Instituto Superior Técnico, 2005.

[70] G. Cybenco. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[71] A. d' Avila Garcez, D. Gabbay, and L. Lamb. Value-based argumentation frameworks as neural-symbolic learning systems. *Journal of Logic and Computation*, 15(6):1041–1058, 2005.

[72] N. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 11:49–510, 1974.

[73] N. da Costa. *Sistemas Formais Inconsistentes.* Cathedra Thesis. UFPR, 1993. In Portuguese, thesis presented in 1963.

[74] N. da Costa and E. Alves. A semantical analysis of the calculi $\mathcal{C}_n$. *Notre Dame Journal of Formal Logic*, 18(4):621–630, 1977.

[75] N. da Costa and W. Carnielli. Paraconsistent deontic logic. *Philosophia - The Philosophical Quarterly of Israel*, 16(3/4):293–305, 1988.

[76] M. D'Agostino and D. Gabbay. Fibred tableaux for multi-implication logics. In *Theorem Proving with Analytic Tableaux and Related Methods*, volume 1071 of *Lecture Notes in Computer Science*, pages 16–35. Springer, 1996.

[77] B. Davey and H. Priestley. *Introduction to Lattices and Order.* Cambridge University Press, 2nd edition, 2002.

[78] A. d'Avila Garcez, K. Broda, and D. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications.* Springer, 2002. Perspectives in Neural Computing.

[79] A. d'Avila Garcez and D. Gabbay. Fibring neural networks. In *Proceedings of 19th National Conference on Artificial Intelligence.* AAAI Press, 2004.

[80] A. d'Avila Garcez, L. Lamb, K. Broda, and D. Gabbay. Applying connectionist modal logics to distributed knowledge representation problems. *International Journal on Artificial Intelligence Tools*, 13(1):115–139, 2004.

[81] A. De Morgan. *"On the syllogism" and other logical writings.* Edited, with an introduction, by Peter Heath. Yale University Press, New Haven, Conn., 1966.

[82] L. Fariñas del Cerro and A. Herzig. Combining classical and intuitionistic logic. In Schulz [234], pages 93–102.

[83] H. Deutsch. The completeness of $\mathcal{S}$. *Studia Logica*, 38(2):137–147, 1979.

[84] H. Deutsch. Paraconsistent analytic implication. *Journal of Philosophical Logic*, 13(1):1–11, 1984.

[85] I. D'Ottaviano and N. da Costa. Sur un problème de Jaśkowski. *Comptes Rendus de l'Academie de Sciences de Paris (A-B)*, 270:1349–1353, 1970.

[86] H.-D. Ehrich, C. Caleiro, A. Sernadas, and G. Denker. Logics for specifying concurrent information systems. In J. Chomicki and G. Saake, editors, *Logic for Databases and Information Systems*, pages 167–198. Kluwer, 1998.

[87] J. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[88] R. Epstein. *The Semantic Foundations of Logic.* Oxford University Press, 2nd edition, 1995. Propositional logics, With the assistance and collaboration of W. Carnielli, I. D'Ottaviano, S. Krajewski and R. Maddux.

[89] R. Epstein. *Propositional Logics: The semantic foundations of logic,* with the assistance and collaboration of W.A. Carnielli, I.M.L. D'Ottaviano, S. Krajewski, and R.D. Maddux, volume 1. Wadsworth-Thomson Learning, 2nd edition, 2000.

[90] R. Epstein and W. Carnielli. *Computability: Computable Functions, Logic, and the Foundations of Mathematics.* Wadsworth/Thomson Learning, 2nd edition, 2000.

[91] R. Fagin and J. Halpern. Reasoning about knowledge and probability. *Journal of the Association for Computing Machinery*, 41(2):340–367, 1994.

[92] R. Fajardo and M. Finger. Non-normal modalisation. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyaschev, editors, *Advances in Modal Logic, volume 4*, pages 83–95. King's College Publications, 2003.

[93] V. Fernández. Society semantics for n-valued logics. Master's thesis, State University of Campinas (UNICAMP), 2001. In Portuguese.

[94] V. Fernández. *Fibring Logics in the Leibniz Hierarchy.* PhD thesis, State University of Campinas (UNICAMP), 2005. In Portuguese.

[95] J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In D. Sannella and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 332 of *Lecture Notes in Computer Science*, pages 44–72. Springer, 1988.

[96] M. Finger and D. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1(3):203–233, 1992.

[97] M. Finger and D. Gabbay. Combining temporal logic systems. *Notre Dame Journal of Formal Logic*, 37(2):204–232, 1996.

[98] M. Finger and M. Weiss. The unrestricted combination of temporal logic systems. *Logic Journal of the IGPL*, 10(2):165–189, 2002.

[99] M. Fitting. Logics with several modal operators. *Theoria*, 35:259–266, 1969.

[100] L. Fu. *Neural Networks in Computer Intelligence*. McGraw Hill, 1994.

[101] K. Gödel. Zum intuitionistischen Aussagenkalküls. *Anz. Akad. Wiss. Wien*, 69:65–66, 1932.

[102] D. Gabbay. Craig interpolation theorem for intuitionistic logic and extensions. III. *The Journal of Symbolic Logic*, 42(2):269–271, 1977.

[103] D. Gabbay, editor. *What is a Logical System?*, volume 4 of *Studies in Logic and Computation*. The Clarendon Press Oxford University Press, 1994.

[104] D. Gabbay. Fibred semantics and the weaving of logics: Part 1. *The Journal of Symbolic Logic*, 61(4):1057–1120, 1996.

[105] D. Gabbay. *Labelled Deductive Systems*, volume 1 of *Oxford Logic Guides*. Oxford University Press, 1996.

[106] D. Gabbay. An overview of fibred semantics and the combination of logics. In Schulz [234], pages 1–55.

[107] D. Gabbay. *Elementary Logics: A Procedural Perspective*. Prentice Hall, 1998.

[108] D. Gabbay. *Fibring Logics*. Oxford University Press, 1999.

[109] D. Gabbay. Dynamics of practical reasoning. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, volume 2*, pages 179–224. CSLI Publications, 2000.

[110] D. Gabbay. Integrating diverse flows of information using lds. Technical report, King's College London, 2002.

[111] D. Gabbay, H. Barringuer, and J. Woods. Network modalities. Technical report, King's College London, 2004.

[112] D. Gabbay and de M. Rijke, editors. *Frontiers of Combining Systems, 2*, volume 7 of *Studies in Logic and Computation*. Research Studies Press Ltd., 2000.

[113] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, North-Holland, 2003.

[114] D. Gabbay and N. Olivetti. *Goal Directed Proof Theory*. Kluwer, 2000.

[115] D. Gabbay and V. Shehtman. Products of modal logics. I. *Logic Journal of the IGPL*, 6(1):73–146, 1998.

[116] D. Gabbay and V. Shehtman. Products of modal logics. III. Products of modal and temporal logics. *Studia Logica*, 72(2):157–183, 2002.

[117] D. Gabbay and J. Woods, editors. *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*. Elsevier, North-Holland, 2005.

[118] D. Gabbay and J. Woods, editors. *The Rise of Modern Logic from Leibniz to Frege*, volume 3 of *Handbook of the History of Logic*. Elsevier, North-Holland, 2005.

[119] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. Combining spatial and temporal logics: expressiveness vs. complexity. *Journal of Artificial Intelligence Research*, 23:167–243 (electronic), 2005.

[120] D. Gabelaia, A. Kurucz, F. Wolter, and M. Zakharyaschev. Products of transitive modal logics. *The Journal of Symbolic Logic*, 70(3):993–1021, 2005.

[121] M. Gardner. *Logic Machines and Diagrams*. University of Chicago Press, 2nd edition, 1982.

[122] G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39(1):176–210, 1935.

[123] G. Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39(1):405–431, 1935.

[124] G. Gentzen. *The Collected Papers of Gerhard Gentzen*. Edited by M. Szabo. Studies in Logic and the Foundations of Mathematics. North Holland, 1969.

[125] J.-Y. Girard. Linear logic: its syntax and semantics. In *Advances in linear logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 1–42. Cambridge University Press, 1995.

[126] K. Gödel. Eine interpretation des intuitionistischen aussagenkalküls. *Ergebnisse eines mathematischen Kolloquiums*, 4:39–40, 1932.

[127] K. Gödel. Zur intuitionistischen arithmetik und zahlentheorie. *Ergebnisse eines mathematischen Kolloquiums*, 4:34–38, 1933.

[128] K. Gödel. Eine Interpretation des intuitionistischen Aussagenkalküls. In S. Feferman, editor, *Collected Works. Vol. I*, pages 300–303. The Clarendon Press Oxford University Press, 1986. Publications 1929–1936.

[129] J. Goguen and R. Burstall. Introducing institutions. In E. Clarke and D. Kozen, editors, *Proceedings Logics of Programming Workshop*, volume 164 of *Lecture Notes in Computer Science*, pages 221–256. Springer, 1984.

[130] J. Goguen and R. Burstall. A study in the foundations of programming methodology: specifications, institutions, charters and parchments. In *Category Theory and Computer Programming*, volume 240 of *Lecture Notes in Computer Science*, pages 313–333. Springer, 1986.

[131] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[132] J. Goguen and J. Meseguer. Completeness of many-sorted equational logic. *Houston Journal of Mathematics*, 11(3):307–334, 1985.

[133] J. Goguen, J. Tatcher, and E. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R. Yeh, editor, *Current Trends in Programming Methodology IV: Data Structuring*, pages 80–144. Prentice-Hall, 1978.

[134] R. Goldblatt. *Mathematics of Modality*, volume 43. CSLI Publications, 1993.

[135] R. Goldblatt. Mathematical modal logic: a view of its evolution. In Gabbay and Woods [117], pages 13–99.

[136] O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[137] S. Gottwald. *A Treatise on Many-valued Logics*, volume 9. Research Studies Press, 2001.

[138] B. Gramlich, editor. *Frontiers of Combining Systems*, volume 3717 of *Lecture Notes in Artificial Intelligence*. Springer, 2005.

[139] E. Gyftodimos and P. Flach. Non-truth-functional fibred semantics. In H. Arabnia, editor, *Proceedings of the International Conference on Artificial Intelligence*, pages 841–847. CSREA Press, 2001.

[140] R. Hähnle. Advanced many-valued logics. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 297–395. Kluwer, 2nd edition, 2001.

[141] J. Halpern and K. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–514, 2005.

[142] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic logic*. Foundations of Computing Series. MIT Press, 2000.

[143] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1999.

[144] L. Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, 14:159–166, 1949.

[145] L. Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15:81–91, 1950.

[146] L. Henkin, J. Monk, and A. Tarski. *Cylindric Algebras. Part I*, volume 64 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, North-Holland, 2005.

[147] T. Henzinger, R. Jhala, R. Majumdar, and K. McMillan. Abstractions from proofs. In *Proceedings of 31st Annual Symposium on Principles of Programming Languages*, pages 232–244. ACM Press, 2004.

[148] D. Hilbert, W. Ackermann, and R. Luce. *Principles of Mathematical Logic*. American Mathematical Society, 2003.

[149] J. Hintikka. *Knowledge and Belief*. Corneli University Press, 1962.

[150] S. Holldobler, Y. Kalinke, and H. Storr. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence Journal, Special Issue on Neural Networks and Structured Knowledge*, 11(1):45–58, 1999.

[151] S. Holldobler and F. Kurfess. CHCL: A connectionist inference system. In B. Fronhofer and G. Wrightson, editors, *Parallelization in Inference Systems*, pages 318–342. Springer, 1992.

[152] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[153] G. Hughes and M. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, 1996.

[154] B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1999.

[155] M. Jaeger. Complex probabilistic modeling with recursive relational bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32:179–220, 2001.

[156] G. Japaridze and D. de Jongh. The logic of provability. In Buss [33], pages 475–546.

[157] E. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630, 1957.

[158] A. Jones and J. Carmo. Deontic logic and contrary-to-duties. In D. Gabbay and F. Guenthene, editors, *Handbook of Philosophical Logic*, volume 8, pages 318–342. Kluwer, 2nd edition, 2002.

[159] B Jónsson and A. Tarski. Boolean algebras with operators. II. *American Journal of Mathematics*, 74:127–162, 1952.

[160] J. Kelley. *General topology*. Springer-Verlag, New York, 1975. Reprint of the 1955 edition [Van Nostrand, Toronto, Ont.], Graduate Texts in Mathematics, No. 27.

[161] B. Kijsirikul, S. Sinthupinyo, and K. Chongkasemwongse. Approximate match of rules using backpropagation neural networks. *Machine Learning*, 43(3):273–299, 2001.

[162] H. Kirchner and C. Ringeissen, editors. *Frontiers of Combining Systems, 3*, volume 1794 of *Lecture Notes in Computer Science*. Springer, 2000.

[163] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 302–313, 1997.

[164] A. Kolmogorov. On the principle of the excluded middle (russian). *Matematicheskij Sbornik. Akademiya Nauk SSSR i Moskovskoe Matematicheskoe Obshchesto*, 32:646 667, 1925. Translation in [268].

[165] B. Konev, R. Kontchakov, F. Wolter, and M. Zakharyaschev. On dynamic topological and metric logics. *Studia Logica*, 82:in print, 2006.

[166] R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. Spatial logic + temporal logic = ? In Aiello et al. [2], pages 497–564.

[167] M. Kracht. *Tools and Techniques in Modal Logic*, volume 142 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, North Holland, 1999.

[168] M. Kracht and O. Kutz. The semantics of modal predicate logic. I. Counterpart-frames. In *Advances in Modal Logic, Vol. 3 (Leipzig, 2000)*, pages 299–320. World Sci. Publ., River Edge, NJ, 2002.

[169] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *The Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[170] P. Kremer. Dynamic topological logic. *The Bulletin of Symbolic Logic*, 3:375–376, 1997.

[171] P. Kremer and G. Mints. Dynamic topological logic. *The Bulletin of Symbolic Logic*, 3:371–372, 1997.

[172] P. Kremer and G. Mints. Dynamic topological logic. *Annals of Pure and Applied Logic*, 131(1-3):133–158, 2005.

[173] P. Kremer, G. Mints, and V. Rybakov. Axiomatizing the next-interior fragment of dynamic topological logic. *The Bulletin of Symbolic Logic*, 3:376–377, 1997.

[174] S. Kripke. Semantical analysis of modal logic. I. Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[175] J. Lambek. Deductive systems and categories. I. Syntactic calculus and residuated categories. *Mathematical Systems Theory*, 2:287–318, 1968.

[176] J. Lambek. Deductive systems and categories. II. Standard constructions and closed categories. In *Category Theory, Homology Theory and their Applications, I*, volume 86 of *Lecture Notes in Mathematics*, pages 76–122. Springer, 1969.

[177] S. Lauritzen and D. Spiegelhalter. Local computation with probabilities in graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society B*, 50(2):157–254, 1988.

[178] E. Lemmon. *An Introduction to Modal Logic.* Basil Blackwell, 1977. The "Lemmon notes", In collaboration with Dana Scott, Edited by Krister Segerberg, American Philosophical Quarterly, Monograph Series, No. 11.

[179] W. Lenzen. Leibniz's logic. In Gabbay and Woods [118], pages 1–83.

[180] R. Lewin, I. Mikenberg, and M. Schwarze. $\mathcal{C}_1$ is not algebraizable. *Notre Dame Journal of Formal Logic*, 32(4):609–611, 1991.

[181] M. Löb. Solution of a problem of Leon Henkin. *The Journal of Symbolic Logic*, 20:115–118, 1955.

[182] A. Loparić and E. Alves. The semantics of the systems $\mathcal{C}_n$ of da Costa. In A. Arruda, N. da Costa, and A. Sette, editors, *Proceedings of the 3rd Brazilian Conference on Mathematical Logic*, pages 161–172. Sociedade Brasileira de Lógica, 1980.

[183] J. Łoś and R. Suszko. Remarks on sentential logics. *Indagationes Mathematicae*, 20:177–183, 1958.

[184] J. Łukasiewicz. O logice trójwartosciowej. *Ruch Filozoficzny*, 5:170–171, 1920. Translated in English as "On three-valued logic" in [27].

[185] J. Łukasiewicz. A system of modal logic. *Journal of Computing Systems*, 1:111–149, 1953.

[186] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 1998.

[187] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic*. Springer, 1994.

[188] D. Makinson and L. van der Torre. Input/output logics. *Journal of Philosophical Logic*, 29(4):383–408, 2000.

[189] D. Makinson and L. van der Torre. What is input/output logic? In *Foundations of the Formal Sciences II*, volume 17 of *Trends in Logic: Studia Logica Library*, pages 163–174. Kluwer, 2003.

[190] L. Maksimova. Craig's theorem in superintuitionistic logics and amalgamable varieties. *Algebra i Logika*, 16(6):643–681, 741, 1977.

[191] L. Maksimova. Interpolation in superintuitionistic predicate logics with equality. *Algebra i Logika*, 36(5):543–561, 600, 1997.

[192] L. Maksimova. Complexity of interpolation and related problems in positive calculi. *The Journal of Symbolic Logic*, 67(1):397–408, 2002.

[193] L. Maksimova and D. Gabbay. *Interpolation and Definability: Modal and Intuitionistuc Logic*. Oxford University Press, 2005.

[194] G. Malinowski. *Many-Valued Logics*, volume 25 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1993.

[195] M. Manzano. *Extensions of first order logic*, volume 19 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.

[196] J. Marcos. Possible-translations semantics. Master's thesis, State University of Campinas (UNICAMP), 1999. In Portuguese.

[197] J. Marcos. Possible-translations semantics. In Carnielli et al. [50], pages 119–128.

[198] N. Martin and S. Pollard. *Closure Spaces and Logic*, volume 369 of *Mathematics and its Applications*. Kluwer, 1996.

[199] M. Marx. Complexity of products of modal logics. *Journal of Logic and Computation*, 9(2):197–214, 1999.

[200] P. Mateus, J. Rasga, and C. Sernadas. Modal sequent calculi labelled with truth values: Cut elimination. *Logic Journal of the IGPL*, 13(2):173–199, 2005.

[201] P. Mateus and A. Sernadas. Weakly complete axiomatization of exogenous quantum propositional logic. *Information and Computation*, 204(5):771–794, 2006.

[202] P. Mateus, A. Sernadas, and C. Sernadas. Exogenous semantics approach to enriching logics. In G. Sica, editor, *Essays on the Foundations of Mathematics and Logic*, volume 1, pages 165–194. Polimetrica, 2005.

[203] P. Mateus, A. Sernadas, C. Sernadas, and L. Viganò. Modal sequent calculi labelled with truth values: Completeness, duality and analyticity. *Logic Journal of the IGPL*, 12(3):227–274, 2004.

[204] K. McMillan. Interpolation and SAT-based model checking. In F. Somenzi and W. Hunt, editors, *Computer Aided Verification*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2003.

[205] D.H. Mellor, editor. *The Facts of Causation*. Routledge, 1995.

[206] E. Mendelson. *Introduction to Mathematical Logic*. Chapman & Hall, 4th edition, 1997.

[207] J. Meseguer. General logics. In H.-D. Ebbinghaus, J. Fernández-Prida, M. Garrido, D. Lascar, and Rodríguez Artalejo, editors, *Logic Colloquium '87*, volume 129 of *Studies in Logic and the Foundations of Mathematics*, pages 275–329. Elsevier, North Holland, 1989.

[208] J. Meseguer. Membership algebra as a logical framework for equational specification. In F. Parisi-Presicce, editor, *Recent Developments in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*, pages 18–61. Springer, 1998.

[209] C. Mortensen. Every quotient algebra for $\mathcal{C}_1$ is trivial. *Notre Dame Journal of Formal Logic*, 21:694–700, 1980.

[210] T. Mossakowski. Using limits of parchments to systematically construct institutions of partial algebras. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications*, volume 1130 of *Lecture Notes in Computer Science*, pages 379–393. Springer, 1996.

[211] T. Mossakowski, A. Tarlecki, and W. Pawlowski. Combining and representing logical systems. In E. Moggi and G. Rosolini, editors, *Category Theory and Computer Science*, volume 1290 of *Lecture Notes in Computer Science*, pages 177–196. Springer, 1997.

[212] T. Mossakowski, A. Tarlecki, and W. Pawlowski. Combining and representing logical systems using model-theoretic parchments. In F. Parisi Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 1998.

[213] D. Mundici. Complexity of Craig's interpolation. *Fundamenta Informaticae*, 5(3-4):261–278, 1982.

[214] D. Mundici. NP and Craig's interpolation theorem. In *Logic Colloquium '82*, volume 112 of *Studies in Logic and Foundations of Mathematics*, pages 345–358. Elsevier, North Holland, 1984.

[215] R. Neapolitan. *Probabilistic Reasoning in Expert Aystems: Theory and Algorithms*. John Wiley, 1990.

[216] M. Neil, N. Fenton, and L. Neilsen. Building large-scale bayesian networks. *The Knowledge Engineering Review*, 15(3):257–284, 2000.

[217] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[218] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[219] J. Peña, J. Lozano, and P. Larrañaga. Learning recursive Bayesian multinets for clustering by means of constructive induction. *Machine Learning*, 47(1):63–90, 2002.

[220] G. Pinkas. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. *Artificial Intelligence*, 77:203–247, 1995.

[221] A. Pitts. Categorical logic. In S. Abramski, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5, pages 39–128. Oxford University Press, 2001.

[222] J. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1):77–105, 1990.

[223] K. Popper. On the theory of deduction, II: The definitions of classical and intuitionist negation. *Indagationes Mathematicae*, 10:111–120, 1948.

[224] L. Puga, N. da Costa, and W. Carnielli. Kantian and non-Kantian logics. *Logique et Analyse*, 31(121/122):3–9, 1988.

[225] J. Rasga. *Fibring Labelled First-order Based Logics*. PhD thesis, IST, Universidade Técnica de Lisboa, 2003.

[226] J. Rasga, A. Sernadas, C. Sernadas, and L. Viganò. Fibring labelled deduction systems. *Journal of Logic and Computation*, 12(3):443–473, 2002.

[227] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*. Polish Scientific Publishers, 2nd edition, 1968.

[228] S. Read. Hugh MacColl and the algebra of strict implication. *Nordic Journal of Philosophical Logic*, 3(1-2):59–83 (1999), 1998. Hugh MacColl and the tradition of logic (Greifswald, 1998).

[229] G. Restall. *An Introduction to Substructural Logics*. Routledge, 2000.

[230] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986.

[231] P. Rusnock and R. George. Bolzano as a logician. In Gabbay and Woods [118], pages 177–205.

[232] V. Rybakov. *Admissibility of Logical Inference Rules*, volume 136 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, North Holland, 1997.

[233] P. Schroeder-Heister. Popper's structuralist theory of logic. In I. Jarvie, K. Milford, and D. Miller, editors, *Karl Popper. A Centenary Assessment, Volume III*, pages 17–36. Ashgate Publishing Company, 2006.

[234] K. Schulz, editor. *Frontiers of Combining Systems*, volume 3 of *Applied Logic Series*. Kluwer, 1996.

[235] K. Segerberg. Two-dimensional modal logic. *Journal of Philosophical Logic*, 2(1):77–96, 1973.

[236] V. Šehtman. Two-dimensional modal logics. *Akademiya Nauk SSSR. Matematicheskie Zametki*, 23(5):759–772, 1978.

[237] A. Sernadas and C. Sernadas. Combining logic systems: Why, how, what for? *CIM Bulletin*, 15:9–14, December 2003.

[238] A. Sernadas, C. Sernadas, and C. Caleiro. Synchronization of logics. *Studia Logica*, 59(2):217–247, 1997.

[239] A. Sernadas, C. Sernadas, and C. Caleiro. Synchronization of logics with mixed rules: Completeness preservation. In M. Johnson, editor, *Algebraic Methodology and Software Technology*, volume 1349 of *Lecture Notes in Computer Science*, pages 465–478. Springer, 1997.

[240] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.

[241] A. Sernadas, C. Sernadas, C. Caleiro, and T. Mossakowski. Categorial fibring of logics with terms and binding operators. In Gabbay and de M. Rijke [112], pages 295–316.

[242] A. Sernadas, C. Sernadas, and A. Zanardo. Fibring modal first-order logics: Completeness preservation. *Logic Journal of the IGPL*, 10(4):413–451, 2002.

[243] C. Sernadas, J. Rasga, and W. Carnielli. Modulated fibring and the collapsing problem. *The Journal of Symbolic Logic*, 67(4):1541–1569, 2002.

[244] A. Sette. On the propositional calculus P^1. *Mathematica Japonicae*, 18:173–180, 1973.

[245] A. Sette and W. Carnielli. Maximal weakly-intuitionistic logics. *Studia Logica*, 55(1):181–203, 1995.

[246] I. Shapirovsky and V. Shehtman. Modal logics of regions and Minkowski spacetime. *Journal of Logic and Computation*, 15(4):559–574, 2005.

[247] L. Shastri. Advances in SHRUTI: A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence Journal, Special Issue on Neural Networks and Structured Knowledge*, 11:79–108, 1999.

[248] P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, 46:159–216, 1990.

[249] P. Smolensky. Grammar-based connectionist approaches to language. *Cognitive Science*, 23:589–613, 2000.

[250] R. Smullyan. *First-order logic*. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 43. Springer, 1968.

[251] R. Smullyan. Uniform Gentzen systems. *The Journal of Symbolic Logic*, 33:549–559, 1968.

[252] R. Solovay. Provability interpretations of modal logic. *Israel Journal of Mathematics*, 25(3-4):287–304, 1976.

[253] R. Sun and F. Alexandre. *Connectionist Symbolic Integration*. Lawrence Erlbaum Associates, 1997.

[254] M. Szabo. Polycategories. *Communications in Algebra*, 3(8):663–689, 1975.

[255] A. Tarlecki. Moving between logical systems. In M. Haveraaen, O. Owe, and O.J. Dahl, editors, *Recent Trends in Data Type Specifications*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer-Verlag, 1996.

[256] A. Tarlecki, R. Burstall, and J. Goguen. Some fundamental algebraic tools for the semantics of computation. III: Indexed categories. *Theoretical Computer Science*, 91(2):239–264, 1991.

[257] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[258] A. Tarski. *Logic, Semantics, Metamathematics. Papers from 1923 to 1938.* Oxford at the Clarendon Press, 1956. Translated by J. H. Woodger.

[259] R. Thomason. Combinations of tense and modality. In *Handbook of Philosophical Logic, Vol. II*, volume 165 of *Synthese Library*, pages 135–165. Reidel, 1984.

[260] D. Touretzky and G. Hinton. A distributed connectionist production system. *Cognitive Science*, 12(3):423–466, 1988.

[261] G. Towell and J. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1):119–165, 1994.

[262] K-E. Tranøy. "Ought" implies "can": A bridge from fact to norm?, Part I. *Ratio*, 14:116–130, 1972.

[263] A. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, second edition, 2000.

[264] J. van Benthem. *Modal Correspondence Theory.* PhD thesis, University of Amsterdam, 1976.

[265] J. van Benthem. Two simple incomplete logics. *Theoria*, 44:25–37, 1978.

[266] J. van Benthem. The variety of consequence, according to Bolzano. *Studia Logica*, 44(4):389–403, 1985.

[267] J. van Benthem. Epistemic logic and epistemology: the state of their affairs. Technical report, Institute for Logic, Language and Computation, 2005.

[268] J. van Heijenoort, editor. *From Frege to Gödel. A Source Book in Mathematical Logic 1879-1931.* Harvard University Press, 1967.

[269] D. Varacca, H. Völtzer, and G. Winskel. Probabilistic event structures and domains. In P. Gardner and N. Yoshida, editors, *CONCUR*, volume 3170 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 2004.

[270] L. Viganò. *Labelled Non-Classical Logics.* Kluwer, 2000.

[271] P. Werbos. *Beyond Regretion: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

[272] P. Werbos. Backpropagation through time: what does it mean and how to do it. In *Proceedings of the IEEE*, volume 78, pages 1550–1560, 1990.

[273] J. Williamson. Foundations for Bayesian networks. In D. Corfield and J. Williamson, editors, *Foundations of Bayesianism*, pages 75–115. Kluwer, 2001. Applied Logic Series.

[274] J. Williamson. Maximising entropy efficiently. *Electronic Transactions in Artificial Intelligence Journal*, 6, 2002.

[275] J. Williamson and D. Gabbay. Recursive causality in bayesian networks and self-fibring. In D. Gillies, editor, *Laws and Models of Science*, pages 173–247. King's College Publications, 2004.

[276] J. Williamson and D. Gabbay. Recursive Bayesian networks and self-fibring logics. In D. Gillies, editor, *Laws and Models in Science*, pages 173–221. King's College Publications, 2005.

[277] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 4, pages 1–148. Oxford University Press, 1995.

[278] R. Wójcicki. Logical matrices strongly adequate for structural sentential calculi. *Bulletin de l'Académie Polonaise des Sciences, Série des Sciences Mathématiques, Astronomiques et Physiques*, 17:333–335, 1969.

[279] R. Wójcicki. Matrix approach in methodology of sentential calculi. *Studia Logica*, 32:7–37, 1973.

[280] R. Wójcicki. *Theory of Logical Calculi: Basic Theory of Consequence Operations*, volume 199 of *Synthese Library*. Kluwer, 1988.

[281] F. Wolter. Fusions of modal logics revisited. In *Advances in Modal Logic, volume 1*, pages 361–379. CSLI Publications, 1998.

[282] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. *The Journal of Symbolic Logic*, 66(1):414–439, 2001.

Subject index

Table of symbols

List of Figures

APPLIED LOGIC SERIES

---

1. D. Walton: *Fallacies Arising from Ambiguity.* 1996 ISBN 0-7923-4100-7
2. H. Wansing (ed.): *Proof Theory of Modal Logic.* 1996 ISBN 0-7923-4120-1
3. F. Baader and K.U. Schulz (eds.): *Frontiers of Combining Systems.* First International Workshop, Munich, March 1996. 1996 ISBN 0-7923-4271-2
4. M. Marx and Y. Venema: *Multi-Dimensional Modal Logic.* 1996 ISBN 0-7923-4345-X
5. S. Akama (ed.): *Logic, Language and Computation.* 1997 ISBN 0-7923-4376-X
6. J. Goubault-Larrecq and I. Mackie: *Proof Theory and Automated Deduction.* 1997 ISBN 0-7923-4593-2
7. M. de Rijke (ed.): *Advances in Intensional Logic*. 1997 ISBN 0-7923-4711-0
8. W. Bibel and P.H. Schmitt (eds.): *Automated Deduction - A Basis for Applications*. Volume I. Foundations - Calculi and Methods. 1998 ISBN 0-7923-5129-0
9. W. Bibel and P.H. Schmitt (eds.): *Automated Deduction - A Basis for Applications.* Volume II. Systems and Implementation Techniques. 1998 ISBN 0-7923-5130-4
10. W. Bibel and P.H. Schmitt (eds.): *Automated Deduction - A Basis for Applications*. Volume III. Applications. 1998 ISBN 0-7923-5131-2
(Set vols. I-III: ISBN 0-7923-5132-0)
11. S.O. Hansson: *A Textbook of Belief Dynamics*. Theory Change and Database Updating. 1999 Hb: ISBN 0-7923-5324-2; Pb: ISBN 0-7923-5327-7
Solutions to exercises. 1999. Pb: ISBN 0-7923-5328-5
Set: (Hb): ISBN 0-7923-5326-9; (Pb): ISBN 0-7923-5329-3
12. R. Pareschi and B. Fronhöfer (eds.): *Dynamic Worlds from the Frame Problem to Knowledge Management*. 1999 ISBN 0-7923-5535-0
13. D.M. Gabbay and H. Wansing (eds.): *What is Negation?* 1999 ISBN 0-7923-5569-5
14. M. Wooldridge and A. Rao (eds.): *Foundations of Rational Agency*. 1999 ISBN 0-7923-5601-2
15. D. Dubois, H. Prade and E.P. Klement (eds.): *Fuzzy Sets, Logics and Reasoning about Knowledge*. 1999 ISBN 0-7923-5911-1
16. H. Barringer, M. Fisher, D. Gabbay and G. Gough (eds.): *Advances in Temporal Logic*. 2000 ISBN 0-7923-6149-0
17. D. Basin, M.D. Agostino, D.M. Gabbay, S. Matthews and L. Viganò (eds.): *Labelled Deduction*. 2000 ISBN 0-7923-6237-3
18. P.A. Flach and A.C. Kakas (eds.): *Abduction and Induction.* Essays on their Relation and Integration. 2000 ISBN 0-7923-6250-0
19. S. Hölldobler (ed.): *Intellectics and Computational Logic*. Papers in Honor of Wolfgang Bibel. 2000 ISBN 0-7923-6261-6

20. P. Bonzon, M. Cavalcanti and Rolf Nossum (eds.): *Formal Aspects of Context.* 2000 ISBN 0-7923-6350-7
21. D.M. Gabbay and N. Olivetti: *Goal-Directed Proof Theory*. 2000 ISBN 0-7923-6473-2
22. M.-A. Williams and H. Rott (eds.): *Frontiers in Belief Revision*. 2001 ISBN 0-7923-7021-X
23. E. Morscher and A. Hieke (eds.): *New Essays in Free Logic*. In Honour of Karel Lambert. 2001 ISBN 1-4020-0216-5
24. D. Corfield and J. Williamson (eds.): *Foundations of Bayesianism*. 2001 ISBN 1-4020-0223-8
25. L. Magnani, N.J. Nersessian and C. Pizzi (eds.): *Logical and Computational Aspects of Model-Based Reasoning*. 2002 Hb: ISBN 1-4020-0712-4; Pb: ISBN 1-4020-0791-4
26. D.J. Pym: *The Semantics and Proof Theory of the Logic of Bunched Implications*. 2002 ISBN 1-4020-0745-0
27. P.B. Andrews: *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Second edition. 2002 ISBN 1-4020-0763-9
28. F.D. Kamareddine: *Thirty Five Years of Automating Mathematics*. 2003 ISBN 1-4020-1656-5
29. F. Kamareddine, T. Laan and R. Nederpelt: *A Modern Perspective on Type Theory*. From its Origins until Today. 2004 ISBN 1-4020-2334-0
30. H. Leitgeb: *Inference on the Low Level*. An Investigation into Deduction, Nonmonotonic Reasoning, and the Philosophy of Cognition. 2004 ISBN 1-4020-2492-4
31. R. Caferra, A. Leitsch and N. Peltier: *Automated Model Building*. 2004 ISBN 1-4020-2652-8
32. J.H. Woods: *The Death of Argument*. Fallacies in Agent-Based Reasoning. 2004 ISBN 1-4020-2663-3
33. M. Thielscher: *Reasoning Robots*. The Art and Science of Programming Robotic Agents. 2005 ISBN 1-4020-3068-1
34. P. Wang: *Rigid Flexibility.* The Logic of Intelligence. 2006 ISBN 978-1-4020-5044-2
35. W. Carnielli, M. Coniglio, D.M. Gabbay, P. Gouveia and C. Sernadas: *Analysis and Synthesis of Logics*. How to Cut and Paste Reasoning Systems. 2008 ISBN 978-1-4020-6781-5