BRICS LS-95-1 J. van Oosten: Basic Category Theory

BRICS

Basic Research in Computer Science

Basic Category Theory

Jaap van Oosten

BRICS Lecture Series LS-95-1

ISSN 1395-2048 January 1995

See back inner page for a list of recent publications in the BRICS Lecture Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@daimi.aau.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.aau.dk/BRICS/`
`ftp ftp.brics.aau.dk (cd pub/BRICS)`

Basic Category Theory

Jaap van Oosten

Jaap van Oosten
BRICS[1]
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark

[1] **B**asic **R**esearch **I**n **C**omputer **S**cience,
Centre of the Danish National Research Foundation.

Preface

These notes contain the material of a short course on categories I gave in Århus in the autumn of 1994, as part of Glynn Winskel's semantics course. Later on, while writing, I added some material, but not much.

The style in which they are written reflects my view on category theory: it is, especially at this low level, practice rather than theory which counts. I have therefore given many proofs as exercises. If you really want to get a grip on the subject, I strongly suggest you do as many of them as you can.

The same goes for the examples. They are the flesh and bones of the theory, and many of them have been chosen so they are a recurring theme; functors $\mathcal{C} \leftrightarrows \mathcal{D}$ may be given as examples in chapter 1, be shown to constitute an adjunction in chapter 5, while this may turn out to be a monadic situation in chapter 6.

For the same reason, references are omitted. Even a sketchy proof, or a hint of the crucial argument, is better than an intimidating reference to [R].

Of course, the examples will be best understood by students who are familiar with the mathematical notions involved, but in general these notes do not require a lot of mathematical background, except for some basic knowledge of groups, rings and topological spaces (although examples on the latter may be skipped, since I have not pursued them through the whole text).

What I *did* presuppose is some familiarity with logic and the λ-calculus. Although definitions are given, standard facts about substitution and the like are suppressed (a teacher can easily supply them when he gives the course). This familiarity does not include a good understanding of set theory or even an inkling of the size problems one can run into. I've used the terms "set" and "small" wherever necessary, although I don't suppose they mean much to many students. For that reason I've also omitted a proof of Freyd's Adjoint Functor Theorem and an explanation of the role of the solution set condition.

Apart from chapters 4 and 7, where in spite of the fact that the results are well-known I haven't been able to find references where they are treated in a concise enough form, and so had to develop the material myself, everything is pretty standard. I have consulted the following sources:

- S. MacLane, *Categories for the Working Mathematician*, Springer (Berlin) 1971.
  Still the best text. For non-mathematicians it may be a little tough going, but it is worth the trouble.

- F. Borceux, *Handbook of Categorical Algebra*, (Encyclopaedia of Mathematics and its Applications) Cambridge University Press (Cambridge) 1994.
  Next best. Gives a lot of material in a very readable style; also on specialized topics. Three volumes.

A strange error in the definition of Grothendieck universes in the first chapter, making the definition inconsistent, supports the point about set theory, I made before.

Many concrete examples. The reader will find many answers to my exercises in this book.

- M. Barr & C. Wells, *Category Theory for Computing Science*, Prentice Hall (New York) 1990.
  At this moment out of print. The emphasis on sketches is debatable, for a first course in the theory. Otherwise a very valuable source.

- P.T. Johnstone, *Stone Spaces*, Cambridge University Press (Cambridge) 1982. Not a book on category theory proper, but a systematic study on various dualities of the Stone type. A lot of material on posetal structures like frames, Boolean algebras etc.

- A. Asperti, *Categorical Topics in Computer Science*, Ph.D. Thesis, Pisa 1990. Later reworked into:
  A. Asperti & G. Longo, *Categories, Types, and Structures: An Introduction to Category Theory for the Working Computer Scientist* (Foundations of Computing), MIT Press (Cambridge Massachusetts) 1991.

- M. Makkai & G. Reyes, *First Order Categorical Logic* (Lecture Notes in Mathematics 611), Springer (Berlin) 1977.
  "The" book on categorical logic. It is my feeling that a sequel is badly needed. The main ideas are developed here.

- S. MacLane & I. Moerdijk, *Sheaves in Geometry and Logic* (Universitext), Springer 1992.
  Treats topos theory, with important applications to logic. Can almost be read from scratch.

- J. Lambek & P. Scott, *Introduction to higher order categorical logic*, Cambridge University Press (Cambridge) 1986.
  This may very well be a book of the future, but for a first acquaintance with category theory the approach is too formal for my taste. Gives a very elaborate account of the correspondences between type theories and certain types of categories.

Of course this list doesn't make any pretense whatsoever at being complete or even a guide into the literature. It mainly reflects my personal attitude.

Acknowledgements. I am grateful to the group of students who patiently and critically sat through my lectures, and in particular to Thomas Hildebrandt and Søren Bøgh Lassen who pointed out mistakes in my original hand-written notes.

The help of my office mate Vladi Sassone, has been invaluable. A critical reading by him of the whole first version revealed a couple of embarassing mistakes ("the functor $(-) \times X$ also has a left adjoint", ha ha—there is no limit to what a confused brain can come up with); then he put a lot of effort in the visual layout of the text, teaching me `emacs` and LaTeX in the process, and designed the rococo painting which is the title page.

It goes without saying that the remaining errors are mine, and that the poor visual quality of the text is a testimony of my ignorance of LaTeX, which I am not proud of.

References

[R] J. Razdajev, *Some facts about functors*, Novosibirsk Journal of Diving Research XLVII (1947), pp. 634-98 (Russian)

Contents

1 Categories and Functors

1.1 Definitions and examples

A *category* $\mathcal{C}$ is given by a class $\mathcal{C}_0$ of *objects* and a class $\mathcal{C}_1$ of *arrows* which have the following structure.

- Each arrow has a *domain* and a *codomain* which are objects; one writes $f : X \to Y$ or $X \xrightarrow{f} Y$ if X is the domain of the arrow f, and Y its codomain. One also writes $X = \mathrm{dom}(f)$ and $Y = \mathrm{cod}(f)$;

- Given two arrows f and g such that $\mathrm{cod}(f) = \mathrm{dom}(g)$, the *composition* of f and g, written gf, is defined and has domain $\mathrm{dom}(f)$ and codomain $\mathrm{cod}(g)$:
$$X \xrightarrow{f} Y \xrightarrow{g} Z$$

- Composition is *associative*, that is: given $f : X \to Y$, $g : Y \to Z$ and $h : Z \to W$, $h(gf) = (hg)f$;

- For every object X there is an *identity* arrow $\mathrm{id_X}$, satisfying $\mathrm{id_X g = g}$ for every $g : Y \to X$ and $f\mathrm{id_X = f}$ for every $f : X \to Y$.

Exercise 1. Show that $\mathrm{id_X}$ is the *unique* arrow with domain X and codomain X with this property.

Instead of "arrow" we also use the terms "morphism" or "map".
Examples

a) **1** is the category with one object $*$ and one arrow, id_*;

b) **0** is the empty category;

c) A *preorder* is a set X together with a binary relation $\leq$ which is reflexive (i.e. $x \leq x$ for all $x \in X$) and transitive (i.e. $x \leq y$ and $y \leq z$ imply $x \leq z$ for all $x, y, z \in X$). This can be viewed as a category, with set of objects X and exactly one arrow: $x \to y$ iff $x \leq y$.

Exercise 2. Prove this. Prove also the converse: if $\mathcal{C}$ is a category such that $\mathcal{C}_0$ is a set, and such that for any two objects X, Y of $\mathcal{C}$ there is at most one arrow: $X \to Y$, then $\mathcal{C}_0$ is a preordered set.

d) A *monoid* is a set X together with a binary operation, written like multiplication: xy for $x, y \in X$, which is associative and has a unit element $e \in X$, satisfying $ex = xe = x$ for all $x \in X$. Such a monoid is a category with one object, and an arrow x for every $x \in X$.

e) Set is the category which has the class of all sets as objects, and functions between sets as arrows.

Most basic categories have as objects certain mathematical structures, and the structure-preserving functions as morphisms. Examples:

f) Top is the category of topological spaces and continuous functions.

g) Grp is the category of groups and group homomorphisms.

h) Rng is the category of rings and ring homomorphisms.

i) Grph is the category of graphs and graph homomorphisms.

j) Pos is the category of partially ordered sets and monotone functions.

Given two categories $\mathcal{C}$ and $\mathcal{D}$, a *functor* $F : \mathcal{C} \to \mathcal{D}$ consists of operations $F_0 : \mathcal{C}_0 \to \mathcal{D}_0$ and $F_1 : \mathcal{C}_1 \to \mathcal{D}_1$, such that for each $f : X \to Y$, $F_1(f) : F_0(X) \to F_0(Y)$ and:

- for $X \xrightarrow{f} Y \xrightarrow{g} Z$, $F_1(gf) = F_1(g)F_1(f)$;
- $F_1(\mathrm{id_X}) = \mathrm{id_{F_0(X)}}$ for each $X \in \mathcal{C}_0$.

But usually we write just F instead of F_0, F_1.
Examples.

a) There is a functor U : Top $\to$ Set which assigns to any topological space X its underlying set. We call this functor "forgetful": it "forgets" the mathematical structure. Similarly, there are forgetful functors Grp $\to$ Set, Grph $\to$ Set, Rng $\to$ Set, Pos $\to$ Set etcetera;

b) For every category $\mathcal{C}$ there is a unique functor $\mathcal{C} \to \mathbf{1}$ and a unique one $\mathbf{0} \to \mathcal{C}$;

c) Given two categories $\mathcal{C}$ and $\mathcal{D}$ we can define the *product category* $\mathcal{C} \times \mathcal{D}$ which has as objects pairs $(C, D) \in \mathcal{C}_0 \times \mathcal{D}_0$, and as arrows:$(C, D) \to (C', D')$ pairs (f, g) with $f : C \to C'$ in $\mathcal{C}$, and $g : D \to D'$ in $\mathcal{D}$. There are functors $\pi_0 : \mathcal{C} \times \mathcal{D} \to \mathcal{C}$ and $\pi_1 : \mathcal{C} \times \mathcal{D} \to \mathcal{D}$;

d) Given two functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$ one can define the composition $GF : \mathcal{C} \to \mathcal{E}$. This composition is of course associative and since we have, for any category $\mathcal{C}$, the *identity functor* $\mathcal{C} \to \mathcal{C}$, we have a category Cat which has categories as objects and functors as morphisms.

e) Given a set A, consider the set $\tilde{A}$ of strings $a_1 \ldots a_n$ on the alphabet $A \cup A^{-1}$ (A^{-1} consists of elements a^{-1} for each element a of A; the sets A and A^{-1} are disjoint and in 1-1 correspondence with each other), such that for no $x \in A$, xx^{-1} or $x^{-1}x$ is a substring of $a_1 \ldots a_n$. Given two such strings $\vec{a} = a_1 \ldots a_n, \vec{b} = b_1 \ldots b_m$, let $\vec{a} \star \vec{b}$ the string formed by first taking $a_1 \ldots a_n b_1 \ldots b_m$ and then removing from this string, successively, substrings of form xx^{-1} or $x^{-1}x$, until one has an element of $\tilde{A}$.

This defines a group structure on $\tilde{A}$. $\tilde{A}$ is called the *free group* on the set A.

Exercise 3. Prove this, and prove that the assignment $A \mapsto \tilde{A}$ is part of a functor: Set $\to$ Grp. This functor is called the *free functor*.

f) Every directed graph can be made into a category as follows: the objects are the vertices of the graph and the arrows are paths in the graph. This defines a functor from the category Dgrph of directed graphs to Cat. The image of a directed graph D under this functor is called the category *generated* by the graph D.

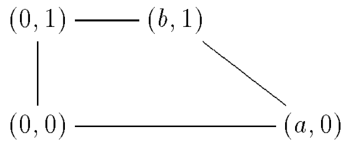
g) **Quotient categories**. Given a category $\mathcal{C}$, a *congruence relation* on $\mathcal{C}$ specifies, for each pair of objects X, Y, an equivalence relation $\sim_{X,Y}$ on the class of arrows $\mathcal{C}(X, Y)$ which have domain X and codomain Y, such that

- for $f, g : X \to Y$ and $h : Y \to Z$, if $f \sim_{X,Y} g$ then $hf \sim_{X,Z} hg$;
- for $f : X \to Y$ and $g, h : Y \to Z$, if $g \sim_{Y,Z} h$ then $gf \sim_{X,Z} hf$.

Given such a congruence relation $\sim$ on $\mathcal{C}$, one can form the quotient category $\mathcal{C}/\sim$ which has the same objects as $\mathcal{C}$, and arrows $X \to Y$ are $\sim_{X,Y}$-equivalence classes of arrows $X \to Y$ in $\mathcal{C}$.

Exercise 4. Show this and show that there is a functor $\mathcal{C} \to \mathcal{C}/\sim$, which takes each arrow of $\mathcal{C}$ to its equivalence class.

h) An example of this is the following ("homotopy"). Given a topological space X and points $x, y \in X$, a *path* from x to y is a continuous mapping f from some closed interval $[0, a]$ to X with $f(0) = x$ and $f(a) = y$. If $f : [0, a] \to X$ is a path from x to y and $g : [0, b] \to X$ is a path from y to z there is a path $gf : [0, a+b] \to X$ (defined by $gf(t) = \begin{cases} f(t) & t \leq a \\ g(t-a) & \text{else} \end{cases}$) from x to z. This makes X into a category, the *path category* of X, and of course this also defines a functor Top $\to$ Cat. Now given paths $f : [0, a] \to X$, $g : [0, b] \to X$, both from x to y, one can define $f \sim_{x,y} g$ if there is a continuous map $F : A \to X$ where A is the area:



in $\mathbb{R}^2$, such that

$$\begin{array}{rll} F(t,0) = & f(t) & \\ F(t,1) = & g(t) & \\ F(0,s) = & x & s \in [0,1] \\ F(s,t) = & y & (s,t) \text{ on the segment } (b,1)-(a,0) \end{array}$$

One can easily show that this is a congruence relation. The quotient of the path category by this congruence relation is a category called the category of *homotopy classes* of paths in X.

i) let $\mathcal{C}$ be a category such that for every pair (X, Y) of objects the class $\mathcal{C}(X, Y)$ of arrows from X to Y is a set (such $\mathcal{C}$ is called *locally small*).

   For any object C of $\mathcal{C}$ then, there is a functor $h_C : \mathcal{C} \to \mathrm{Set}$ which assigns to any object C' the set $\mathcal{C}(C, C')$. Any arrow $f : C' \to C''$ gives by composition a function $\mathcal{C}(C, C') \to \mathcal{C}(C, C'')$, so we have a functor. A functor of this form is called a *representable functor*.

j) Let $\mathcal{C}$ be a category and C an object of $\mathcal{C}$. The *slice category* $\mathcal{C}/C$ has as objects all arrows g which have codomain C. An arrow from $g : D \to C$ to $h : E \to C$ in $\mathcal{C}/C$ is an arrow $k : D \to E$ in $\mathcal{C}$ such that $hk = g$. Draw like:

$$\begin{tikzcd} D \arrow[rr, "k"] \arrow[dr, "g"'] & & E \arrow[dl, "h"] \\ & C & \end{tikzcd}$$

   We say that *this diagram commutes* if we mean that $hk = g$.

Exercise 5. Convince yourself that the assignment $C \mapsto \mathcal{C}/C$ gives rise to a functor $\mathcal{C} \to \mathrm{Cat}$.

k) Remember that for every group $(G, \cdot)$ we can form a group $(G, \star)$ by putting $f \star g = g \cdot f$.

   For categories the same construction is available: given $\mathcal{C}$ we can form a category $\mathcal{C}^{\mathrm{op}}$ which has the same objects and arrows as $\mathcal{C}$, but with reversed direction; so if $f : X \to Y$ in $\mathcal{C}$ then $f : Y \to X$ in $\mathcal{C}^{\mathrm{op}}$. To make it notationally clear, write $\bar{f}$ for the arrow $Y \to X$ corresponding to $f : X \to Y$ in $\mathcal{C}$. Composition in $\mathcal{C}^{\text{op}}$ is defined by:

$$\bar{f}\bar{g} = \overline{gf}$$

Often one reads the term "contravariant functor" in the literature. What I call functor, is then called "covariant functor". A contravariant functor F from $\mathcal{C}$ to $\mathcal{D}$ inverts the direction of the arrows, so $F_1(f) : F_0(\text{cod}(f)) \to F_0(\text{dom}(f))$ for arrows f in $\mathcal{C}$. In other words, a contravariant functor from $\mathcal{C}$ to $\mathcal{D}$ is a functor from $\mathcal{C}^{\text{op}} \to \mathcal{D}$ (equivalently, from $\mathcal{C}$ to $\mathcal{D}^{\text{op}}$).

Exercise 6. Let $\mathcal{C}$ be locally small. Show that there is a functor (the "Hom functor") $\mathcal{C}(-,-) : \mathcal{C}^{\text{op}} \times \mathcal{C} \to \text{Set}$, assigning to the pair (A, B) of objects of $\mathcal{C}$, the set $\mathcal{C}(A, B)$.

l) Given a partially ordered set $(X, \leq)$ we make a topological space by defining $U \subseteq X$ to be open iff for all $x, y \in X$, $x \leq y$ and $x \in U$ imply $y \in U$ (U is "upwards closed", or an "upper set"). This is a topology, called the *Alexandroff topology* w.r.t. the order $\leq$.

If $(X, \leq)$ and $(Y, \leq)$ are two partially ordered sets, a function $f : X \to Y$ is monotone for the orderings if and only if f is continuous for the Alexandroff topologies. This gives an important functor: Pos $\to$ Top.

Exercise 7. Show that the construction of the quotient category in example g) generalizes that of a quotient group by a normal subgroup. That is, regard a group G as a category with one object; show that there is a bijection between congruence relations on G and normal subgroups of G, and that for a normal subgroup N of G, the quotient category by the congruence relation corresponding to N, is to the quotient group G/N.

m) "Abelianization". Let Abgp be the category of abelian groups and homomorphisms. For every group G the subgroup $[G, G]$ generated by all elements of form $aba^{-1}b^{-1}$ is a normal subgroup. $G/[G, G]$ is abelian, and for every group homomorphism $\phi : G \to H$ with H abelian, there is a unique homomorphism $\bar{\phi} : G/[G, G] \to H$ such that the diagram

$$\begin{tikzcd} & G \arrow[dl, "p"'] \arrow[dr, "\phi"] & \\ G/[G,G] \arrow[rr, "\bar{\phi}"'] & & H \end{tikzcd}$$

commutes. Show that this gives a functor: Grp $\to$ Abgp.

n) "Specialization ordering". Given a topological space X, you can define an ordering $\leq_s$ on X as follows: say $x \leq_s y$ if for all open sets U, if $x \in U$ then $y \in U$.

   For many spaces, $\leq_s$ is trivial (in particular when X is T_1) but in case X is for example the Alexandroff topology on a poset $(X, \leq)$ as in l), then $x \leq_s y$ iff $x \leq y$.

Exercise 8. If $f : X \to Y$ is a continuous map of topological spaces then f is monotone w.r.t. the specialization orderings $\leq_s$. This defines a functor Top $\to$ Pos.

1.2 Some special objects and arrows

We call an arrow $f : A \to B$ *mono* (or a monomorphism, or monomorphic) in a category $\mathcal{C}$, if for any other object C and for any pair of arrows $g, h : C \to A$, $fg = fh$ implies $g = h$.

In Set, f is mono iff f is an injective function. The same is true for Grp, Grph, Rng, Preord, Pos,...

We call an arrow $f : A \to B$ *epi* (epimorphism, epimorphic) if for any pair $g, h : B \to C$, $gf = hf$ implies $g = h$.

The definition of epi is "dual" to the definition of mono. That is, f is epi in the category $\mathcal{C}$ if and only if f is mono in $\mathcal{C}^{\mathrm{op}}$, and vice versa. In general, given a property P of an object, arrow, diagram,...we can associate with P the dual property P^{op}: the object or arrow has property P^{op} in $\mathcal{C}$ iff it has P in $\mathcal{C}^{\mathrm{op}}$.

The *duality principle*, a very important, albeit trivial, principle in category theory, says that any valid statement about categories, involving the properties $P_1, \ldots, P_n$ implies the "dualized" statement (where direction of arrows is reversed) with the P_i replaced by P_i^{op}.

Example. If gf is mono, then f is mono. From this, "by duality", if fg is epi, then f is epi.

Exercise 9. Prove these statements.

In Set, f is epi iff f is a surjective function. This holds (less trivially!) also for Grp, but not for Mon, the category of monoids and monoid homomorphisms:

In Mon, the embedding $\mathbb{N} \to \mathbb{Z}$ is an epimorphism.

For suppose $\mathbb{Z} \underset{g}{\overset{f}{\rightrightarrows}} (M, e, \star)$ two monoid homomorphisms which agree on the nonnegative integers. Then

$$f(-1) = f(-1) \star g(1) \star g(-1) = f(-1) \star f(1) \star g(-1) = g(-1)$$

so f and g agree on the whole of $\mathbb{Z}$.

We say a functor F *preserves* a property P if whenever an object or arrow (or...) has P, its F-image does so.

Now a functor does not in general preserve monos or epis: the example of Mon shows that the forgetful functor Mon $\to$ Set does not preserve epis.

An epi $f : A \to B$ is called *split* if there is $g : B \to A$ such that $fg = \mathrm{id}_B$ (other names: in this case g is called a *section* of f, and f a *retraction* of g).

Exercise 10. By duality, define what a split mono is. Prove that every functor preserves split epis and monos.

$f : A \to B$ is an *isomorphism* if there is $g : B \to A$ such that $fg = \mathrm{id}_B$ and $gf = \mathrm{id}_A$ We call g the *inverse* of f (and vice versa, of course); it is unique if it exists. We also write $g = f^{-1}$.

Every functor preserves isomorphisms.

Exercise 11. In Set, every arrow which is both epi and mono is an isomorphism. Not so in Mon, as we have seen. Here's another one: let CRng1 be the category of commutative rings with 1, and ring homomorphisms (preserving 1) as arrows. Show that the embedding $\mathbb{Z} \to \mathbb{Q}$ is epi in CRng1.

Exercise 12.

i) If two of f, g and fg are iso, then so is the third;

ii) if f is epi and split mono, it is iso;

iii) if f is split epi and mono, f is iso.

A functor F *reflects* a property P if whenever the F-image of something (object, arrow,...) has P, then that something has.

A functor $F : \mathcal{C} \to \mathcal{D}$ is called *full* if for every two objects A, B of $\mathcal{C}$, $F : \mathcal{C}(A, B) \to \mathcal{D}(FA, FB)$ is a surjection. F is *faithful* if this map is always injective.

Exercise 13. A faithful functor reflects epis and monos.

An object X is called *terminal* if for any other object Y there is exactly one morphism $Y \to X$ in the category. Dually, X is *initial* if for all Y there is exactly one $X \to Y$.

Exercise 14. A full and faithful functor reflects the property of being a terminal (or initial) object.

Exercise 15. If X and X' are two terminal objects, they are *isomorphic*, that is there exists an isomorphism between them. Same for initial objects.

2 Natural transformations

2.1 The Yoneda lemma

A *natural transformation* between two functors $F, G : \mathcal{C} \to \mathcal{D}$ consists of a family of morphisms $(\mu_C : FC \to GC)_{C \in \mathcal{C}_0}$ indexed by the collection of objects of $\mathcal{C}$, satisfying the following requirement: for every morphism $f : C \to C'$ in $\mathcal{C}$, the diagram

$$\begin{tikzcd} FC \arrow[r, "\mu_C"] \arrow[d, "Ff"'] & GC \arrow[d, "Gf"] \\ FC' \arrow[r, "\mu_{C'}"'] & GC' \end{tikzcd}$$

commutes in $\mathcal{D}$ (the diagram above is called the naturality square). We say $\mu = (\mu_C)_{C \in \mathcal{C}_0} : F \Rightarrow G$ and we call μ_C the *component at* C of the natural transformation μ.

Given natural transformations $\mu : F \Rightarrow G$ and $\nu : G \Rightarrow H$ we have a natural transformation $\nu\mu = (\nu_C \mu_C)_C : F \Rightarrow H$, and with this composition there is a category $\mathcal{D}^{\mathcal{C}}$ with functors $F : \mathcal{C} \to \mathcal{D}$ as objects, and natural transformations as arrows.

One of the points of the naturality square condition in the definition of a natural transformation is given by the following proposition. Compare with the situation in Set: denoting the set of all functions from X to Y by Y^X, for any set Z there is a bijection between functions $Z \to Y^X$ and functions $Z \times X \to Y$ (Set is *cartesian closed*: see chapter 7).

Proposition 2.1 *For categories $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$ there is a bijection:*

$$\mathrm{Cat}(\mathcal{E} \times \mathcal{C}, \mathcal{D}) \xrightarrow{\sim} \mathrm{Cat}(\mathcal{E}, \mathcal{D}^{\mathcal{C}})$$

Proof. Given $F : \mathcal{E} \times \mathcal{C} \to \mathcal{D}$ define for every object E of $\mathcal{E}$ the functor $F_E : \mathcal{C} \to \mathcal{D}$ by $F_E(C) = F(E, C)$; for $f : C \to C'$ let $F_E(f) = F(\mathrm{id_E}, \mathrm{f}) : \mathrm{F_E(C) = F(E, C) \to F(E, C') = F_E(C')}$

Given $g : E \to E'$ in $\mathcal{E}$, the family $(F(g, \mathrm{id_C}) : \mathrm{F_E(C) \to F_{E'}(C)})_{\mathrm{C} \in \mathcal{C}_0}$ is a natural transformation: $F_E \Rightarrow F_{E'}$. So we have a functor $F \mapsto F_{(-)} : \mathcal{E} \to \mathcal{D}^{\mathcal{C}}$.

Conversely, given a functor $G : \mathcal{E} \to \mathcal{D}^{\mathcal{C}}$ we define a functor $\tilde{G} : \mathcal{E} \times \mathcal{C} \to \mathcal{D}$ on objects by $\tilde{G}(E, C) = G(E)(C)$, and on arrows by $\tilde{G}(g, f) = G(g)_{C'} G(E)(f) =$

$G(E')(f)G(g)_C$:

$$\begin{tikzcd} G(E)(C) = \tilde{G}(E,C) \arrow[r, "G(g)_C"] \arrow[d, "G(E)(f)"'] & \tilde{G}(E',C) = G(E')(C) \arrow[d, "G(E')(f)"] \\ \tilde{G}(E,C') \arrow[r, "G(g)_{C'}"'] & \tilde{G}(E',C') = G(E')(C') \end{tikzcd}$$

Exercise 16. Write out the details. Check that $\tilde{G}$ as just defined, is a functor, and that the two operations

$$\mathrm{Cat}(\mathcal{E} \times \mathcal{C}, \mathcal{D}) \rightleftarrows \mathrm{Cat}(\mathcal{E}, \mathcal{D}^{\mathcal{C}})$$

are inverse to each other. ∎

An important example of natural transformations arises from the functors $h_C : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$ (see example i) in the preceding chapter); defined on objects by $h_C(C') = \mathcal{C}(C', C)$ and on arrows $f : C'' \to C'$ so that $h_C(f)$ is composition with f: $\mathcal{C}(C', C) \to \mathcal{C}(C'', C)$.

Given $g : C_1 \to C_2$ there is a natural transformation

$$h_g : h_{C_1} \Rightarrow h_{C_2}$$

whose components are composition with g.

Exercise 17. Spell this out.

We have, in other words, a functor

$$h_{(-)} : \mathcal{C} \to \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$$

This functor is also often denoted by Y and listens to the name *Yoneda embedding.*

An embedding is a functor which is full and faithful and injective on objects. That Y is injective on objects is easy to see, because $\mathrm{id_C} \in \mathrm{h_C(C)}$ for each object C, and $\mathrm{id_C}$ is in no other set $h_D(E)$; that Y is full and faithful follows from the famous

Proposition 2.2 (Yoneda lemma) *For every object F of $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$ and every object C of $\mathcal{C}$, there is a bijection $f_{C,F} : \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C, F) \to F(C)$. Moreover, this bijection is* natural *in C and F in the following sense: given $g : C' \to C$ in $\mathcal{C}$*

and $\mu : F \Rightarrow F'$ *in* $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$*, the diagram*

$$\begin{tikzcd} \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C, F) \arrow[r, "f_{C,F}"] \arrow[d, "\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(g,\mu)"'] & F(C) \arrow[d, "\mu_{C'}F(g)=F'(g)\mu_C"] \\ \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_{C'}, F') \arrow[r, "f_{C',F'}"'] & F'(C') \end{tikzcd}$$

commutes in Set.

Proof. For every object C' of $\mathcal{C}$, every element f of $h_C(C') = \mathcal{C}(C', C)$ is equal to $\mathrm{id_C f}$ which is $h_C(f)(\mathrm{id_C})$.

If $\kappa = (\kappa_{C'} | C' \in \mathcal{C}_0)$ is a natural transformation: $h_C \Rightarrow F$ then, $\kappa_{C'}(f)$ must be equal to $F(f)(\kappa_C(\mathrm{id_C}))$. So κ is completely determined by $\kappa_C(\mathrm{id_C}) \in$ F(C) and conversely, any element of $F(C)$ determines a natural transformation $h_C \Rightarrow F$.

Given $g : C' \to C$ in $\mathcal{C}$ and $\mu : F \Rightarrow F'$ in $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$, the map $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(g, \mu)$ sends the natural transformation $\kappa = (\kappa_{C''} | C'' \in \mathcal{C}_0) : h_C \Rightarrow F$ to $\lambda = (\lambda_{C''} | C'' \in \mathcal{C}_0)$ where $\lambda_{C''} : h_{C'}(C'') \to F'(C'')$ is defined by

$$\lambda_{C''}(h : C'' \to C') = \mu_{C''}(\kappa_{C''}(gh))$$

Now

$$\begin{array}{rcl} f_{C',F'}(\lambda) & = & \lambda_{C'}(\mathrm{id}_{C'}) \\ & = & \mu_{C'}(\kappa_{C'}(g)) \\ & = & \mu_{C'}(F(g)(\kappa_C(\mathrm{id_C}))) \\ & = & (\mu_{C'}F(g))(f_{C,F}(\kappa)) \end{array}$$

which proves the naturality statement. ∎

Corollary 2.3 *The functor* $Y : \mathcal{C} \to \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$ *is full and faithful.*

Proof. Immediate by the Yoneda lemma, since

$$\mathcal{C}(C, C') \cong h_{C'}(C) \cong \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C, h_{C'})$$

and this bijection is induced by Y. ∎

The use of the Yoneda lemma is often the following. One wants to prove that objects A and B of $\mathcal{C}$ are isomorphic. Suppose one can show that for every object X of $\mathcal{C}$ there is a bijection $f_X : \mathcal{C}(X, A) \to \mathcal{C}(X, B)$ which is natural in

X; i.e. given $g : X' \to X$ in $\mathcal{C}$ one has that

$$\begin{tikzcd}
\mathcal{C}(X,A) \arrow[r, "f_X"] \arrow[d, "\mathcal{C}(g,\mathrm{id_A})"'] & \mathcal{C}(X,B) \arrow[d, "\mathcal{C}(g,\mathrm{id_B})"] \\
\mathcal{C}(X',A) \arrow[r, "f_{X'}"'] & \mathcal{C}(X',B)
\end{tikzcd}$$

commutes.

Then one can conclude that A and B are isomorphic in $\mathcal{C}$; for, from what one has just shown it follows that h_A and h_B are isomorphic objects in $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$; that is, $Y(A)$ and $Y(B)$ are isomorphic. Since Y is full and faithful, A and B are isomorphic by the following exercise:

Exercise 18. Check: if $F : \mathcal{C} \to \mathcal{D}$ is full and faithful, and $F(A)$ is isomorphic to $F(B)$ in $\mathcal{D}$, then A is isomorphic to B in $\mathcal{C}$.

Exercise 19. Suppose objects A and B are such that for every object X in $\mathcal{C}$ there is a bijection $f_X : \mathcal{C}(A,X) \to \mathcal{C}(B,X)$, naturally in a sense you define yourself. Conclude that A and B are isomorphic (hint: duality + the previous).

This argument can be carried further. Suppose one wants to show that two functors $F, G : \mathcal{C} \to \mathcal{D}$ are isomorphic as objects of $\mathcal{D}^{\mathcal{C}}$. Let's first spell out what this means:

Exercise 20. Show that F and G are isomorphic in $\mathcal{D}^{\mathcal{C}}$ if and only if there is a natural transformation $\mu : F \Rightarrow G$ such that all components μ_C are isomorphisms (in particular, if μ is such, the family $((\mu_C)^{-1} | C \in \mathcal{C}_0)$ is also a natural transformation $G \Rightarrow F$).

Now suppose one has for each $C \in \mathcal{C}_0$ and $D \in \mathcal{D}_0$ a bijection

$$\mathcal{D}(D, FC) \cong \mathcal{D}(D, GC)$$

natural in D and C. This means that the objects h_{FC} and h_{GC} of $\mathrm{Set}^{\mathcal{D}^{\mathrm{op}}}$ are isomorphic, by isomorphisms which are natural in C. By full and faithfulness of Y, FC and GC are isomorphic in $\mathcal{D}$ by isomorphisms natural in C; which says exactly that F and G are isomorphic as objects of $\mathcal{D}^{\mathcal{C}}$.

2.2 Examples of natural transformations

a) Let M and N be two monoids, regarded as categories with one object as in chapter 1. A functor $F : M \to N$ is then just the same as a homomorphism of monoids. Given two such, say $F, G : M \to N$, a natural transformation $F \Rightarrow G$ is (given by) an element n of N such that $nF(x) = G(x)n$ for all $x \in M$;

b) Let P and Q two preorders, regarded as categories. A functor $P \to Q$ is a monotone function, and there exists a unique natural transformation between two such, $F \Rightarrow G$, exactly if $F(x) \leq G(x)$ for all $x \in P$.

Exercise 21. In fact, show that if $\mathcal{D}$ is a preorder and the category $\mathcal{C}$ is *small*, i.e. the classes $\mathcal{C}_0$ and $\mathcal{C}_1$ are sets, then the functor category $\mathcal{D}^{\mathcal{C}}$ is a preorder.

c) Let U : Grp $\to$ Set denote the forgetful functor, and F : Set $\to$ Grp the free functor (see chapter 1). There are natural transformations $\varepsilon : FU \Rightarrow \mathrm{id}_{\mathrm{Grp}}$ and $\eta : \mathrm{id}_{\mathrm{Set}} \Rightarrow \mathrm{UF}$.

Given a group G, ε_G takes the string $\sigma = g_1 \ldots g_n$ to the product $g_1 \cdots g_n$ (here, the "formal inverses" g_i^{-1} are interpreted as the real inverses in G!).

Given a set A, $\eta_A(a)$ is the singleton string a.

d) Let i : Abgp $\to$ Grp be the inclusion functor and r : Grp $\to$ Abgp the abelianization functor defined in example m) in chapter 1. There is $\varepsilon : ri \Rightarrow \mathrm{id}_{\mathrm{Abgp}}$ and $\eta : \mathrm{id}_{\mathrm{Grp}} \Rightarrow \mathrm{ir}$.

The components η_G of η are the quotient maps $G \to G/[G, G]$; the components of ε are isomorphisms.

e) There are at least two ways to associate a category to a set X: let $F(X)$ be the category with as objects the elements of X, and as only arrows identities (a category of the form $F(X)$ is called *discrete*; and $G(X)$ the category with the same objects but with exactly one arrow $f_{x,y} : x \to y$ for each pair (x, y) of elements of X (We might call $G(X)$ an *indiscrete category*).

Exercise 22. Check that F and G can be extended to functors: Set $\to$ Cat and describe the natural transformation $\mu : F \Rightarrow G$ which has, at each component, the identity function on objects.

f) Every class of arrows of a category $\mathcal{C}$ can be viewed as a natural transformation. Suppose $S \subseteq \mathcal{C}_1$. Let $F(S)$ be the discrete category on S as in the preceding example. There are the two functors $\mathrm{dom}, \mathrm{cod} : F(S) \to \mathcal{C}$, giving the domain and the codomain, respectively. For every $f \in S$ we have $f : \mathrm{dom}(f) \to \mathrm{cod}(f)$, and the family $(f | f \in S)$ defines a natural transformation: $\mathrm{dom} \Rightarrow \mathrm{cod}$.

g) Let A and B be sets. There are functors $(-) \times A$: Set $\to$ Set and $(-) \times B$: Set $\to$ Set. Any function $f : A \to B$ gives a natural transformation $(-) \times A \Rightarrow (-) \times B$.

h) A category $\mathcal{C}$ is called a *groupoid* if every arrow of $\mathcal{C}$ is an isomorphism. Let $\mathcal{C}$ be a groupoid, and suppose we are given, for each object X of $\mathcal{C}$, an arrow μ_X in $\mathcal{C}$ with domain X.

Exercise 23. Show that there is a functor $F : \mathcal{C} \to \mathcal{C}$ in this case, which acts on objects by $F(X) = \mathrm{cod}(\mu_X)$, and that $\mu = (\mu_X | X \in \mathcal{C}_0)$ is a natural transformation: $\mathrm{id}_{\mathcal{C}} \Rightarrow \mathrm{F}$.

i) Given categories $\mathcal{C}$, $\mathcal{D}$ and an object D of $\mathcal{D}$, there is the constant functor $\Delta_D : \mathcal{C} \to \mathcal{D}$ which assigns D to every object of $\mathcal{C}$ and $\mathrm{id_D}$ to every arrow of $\mathcal{C}$.

   Every arrow $f : D \to D'$ gives a natural transformation $\Delta_f : \Delta_D \Rightarrow \Delta_{D'}$ defined by $(\Delta_f)_C = f$ for each $C \in \mathcal{C}_0$.

j) Let $\mathcal{P}(X)$ denote the power set of a set X: the set of subsets of X. The powerset operation can be extended to a functor $\mathcal{P} : \mathrm{Set} \to \mathrm{Set}$. Given a function $f : X \to Y$ define $\mathcal{P}(f)$ by $\mathcal{P}(f)(A) = f[A]$, the image of $A \subseteq X$ under f.

   There is a natural transformation $\eta : \mathrm{id}_{\mathrm{Set}} \Rightarrow \mathcal{P}$ such that $\eta_X(x) = \{x\} \in \mathcal{P}(X)$ for each set X.

   There is also a natural transformation $\mu : \mathcal{P}\mathcal{P} \Rightarrow \mathcal{P}$. Given $A \in \mathcal{P}\mathcal{P}(X)$, so A is a set of subsets of X, we take its union $\bigcup(A)$ which is a subset of X. Put $\mu_X(A) = \bigcup(A)$.

2.3 Equivalence of categories; an example

As will become clear in the following chapters, equality between objects plays only a minor role in category theory. The most important categorical notions are only defined "up to isomorphism". This is in accordance with mathematical practice and with common sense: just renaming all elements of a group does not give you really another group.

We have already seen one example of this: the property of being a terminal object defines an object up to isomorphism. That is, any two terminal objects are isomorphic. There is, in the language of categories, no way of distinguishing between two isomorphic objects, so this is as far as we can get.

However, once we also consider functor categories, it turns out that there is another relation of "sameness" between categories, weaker than isomorphism of categories, and yet preserving all "good" categorical properties. Isomorphism of categories $\mathcal{C}$ and $\mathcal{D}$ requires the existence of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that $FG = \mathrm{id}_{\mathcal{D}}$ and $GF = \mathrm{id}_{\mathcal{C}}$; but bearing in mind that we can't really say meaningful things about equality between objects, we may relax the requirement by just asking that FG is *isomorphic* to $\mathrm{id}_{\mathcal{D}}$ in the functor category $\mathcal{D}^{\mathcal{D}}$ (and the same for GF); doing this we arrive at the notion of *equivalence of categories*, which is generally regarded as the proper notion of sameness.

So two categories $\mathcal{C}$ and $\mathcal{D}$ are *equivalent* if there are functors $F : \mathcal{C} \to \mathcal{D}$, $G : \mathcal{D} \to \mathcal{C}$ and natural transformations $\mu : \mathrm{id}_{\mathcal{C}} \Rightarrow \mathrm{GF}$ and $\nu : \mathrm{id}_{\mathcal{D}} \Rightarrow \mathrm{FG}$ whose components are all isomorphisms. F and G are called *pseudo inverses* of each other. A functor which has a pseudo inverse is also called an *equivalence of categories*.

Exercise 24. Show that a category is equivalent to a discrete category if and only if it is a groupoid and a preorder.

In this section I want to give an important example of an equivalence of categories: the so-called "Lindenbaum-Tarski duality between Set and Complete Atomic Boolean Algebras". A duality between categories $\mathcal{C}$ and $\mathcal{D}$ is an equivalence between $\mathcal{C}^{\mathrm{op}}$ and $\mathcal{D}$ (equivalently, between $\mathcal{C}$ and $\mathcal{D}^{\mathrm{op}}$).

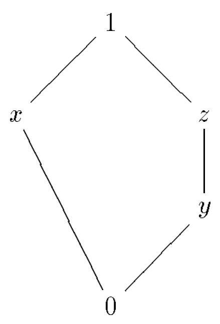
We need some definitions. A *lattice* is a partially ordered set in which every two elements x, y have a least upper bound (or join) $x \vee y$ and a greatest lower bound (or meet) $x \wedge y$; moreover, there exist a least element 0 and a greatest element 1.

Such a lattice is called a *Boolean algebra* if every element x has a *complement* $\neg x$, that is, satisfying $x \vee \neg x = 1$ and $x \wedge \neg x = 0$; and the lattice is *distributive*, which means that $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ for all x, y, z.

In a Boolean algebra, complements are unique, for if both y and z are complements of x, then

$$y = y \wedge 1 = y \wedge (x \vee z) = (y \wedge x) \vee (y \wedge z) = 0 \vee (y \wedge z) = y \wedge z$$

so $y \leq z$; similarly, $z \leq y$ so $y = z$. This is a non-example:



It is a lattice, and every element has a complement, but it is not distributive (check!).

A Boolean algebra B is *complete* if every subset A of B has a least upper bound $\bigvee A$ and a greatest lower bound $\bigwedge A$.

An *atom* in a Boolean algebra is an element x such that $0 < x$ but for no y we have $0 < y < x$. A Boolean algebra is atomic if every x is the join of the atoms below it:

$$x = \bigvee\{a|a \leq x \text{ and } a \text{ is an atom}\}$$

The category CABool is defined as follows: the objects are complete atomic Boolean algebras, and the arrows are complete homomorphisms, that is: $f : B \to C$ is a complete homomorphism if for every $A \subseteq B$,

$$f(\bigvee A) = \bigvee\{f(a)|a \in A\} \text{ and } f(\bigwedge A) = \bigwedge\{f(a)|a \in A\}$$

Exercise 25. Show that $1 = \bigwedge\emptyset$ and $0 = \bigvee\emptyset$. Conclude that a complete homomorphism preserves 1, 0 and complements.

Exercise 26. Show that $\bigwedge A = \neg\bigvee\{\neg a|a \in A\}$ and conclude that if a function preserves all $\bigvee$'s, 1 and complements, it is a complete homomorphism.

Theorem 2.4 *The category* CABool *is equivalent to* $\mathrm{Set}^{\mathrm{op}}$*.*

Proof. For every set X, $\mathcal{P}(X)$ is a complete atomic Boolean algebra and if $f : Y \to X$ is a function, then $f^{-1} : \mathcal{P}(X) \to \mathcal{P}(Y)$ which takes, for each subset of X, its inverse image under f, is a complete homomorphism. So this defines a functor $F : \mathrm{Set}^{\mathrm{op}} \to \mathrm{CABool}$.

Conversely, given a complete atomic Boolean algebra B, let $G(B)$ be the set of atoms of B. Given a complete homomorphism $g : B \to C$ we have a function $G(g) : G(C) \to G(B)$ defined by: $G(g)(c)$ is the unique $b \in G(B)$ such that $c \leq g(b)$. This is well-defined: first, there is an atom b with $c \leq g(b)$ because $1_B = \bigvee G(B)$ (B is atomic), so $1_C = g(1_B) = \bigvee\{g(b)|b \text{ is an atom}\}$ and:

Exercise 27. Prove: if c is an atom and $c \leq \bigvee A$, then there is $a \in A$ with $c \leq a$ (hint: prove for all a, b that $a \wedge b = 0 \Leftrightarrow a \leq \neg b$, and prove for a, c with c atom: $c \not\leq a \Leftrightarrow a \leq \neg c$).

Secondly, the atom b is unique since $c \leq g(b)$ and $c \leq g(b')$ means $c \leq g(b) \wedge g(b') = g(b \wedge b') = g(0) = 0$.

So we have a functor $G : \mathrm{CABool} \to \mathrm{Set}^{\mathrm{op}}$.

Now the atoms of the Boolean algebra $\mathcal{P}(X)$ are exactly the singleton subsets of X, so $GF(X) = \{\{x\}|x \in X\}$ which is clearly isomorphic to X. On the other hand, $FG(B) = \mathcal{P}(\{b \in B|b \text{ is an atom}\})$. There is a map from $FG(B)$ to B which sends each set of atoms to its least upper bound in B, and this map is an isomorphism in CABool. ∎

Exercise 28. Prove the last statement: that the map from $FG(B)$ to B, defined in the last paragraph of the proof, is an isomorphism.

3 (Co)cones and (co)limits

3.1 Limits

Given a functor $F : \mathcal{C} \to \mathcal{D}$, a *cone* for F consists of an object D of $\mathcal{D}$ together with a natural transformation $\mu : \Delta_D \Rightarrow F$ (Δ_D is the constant functor with value D). In other words, we have a family $(\mu_C : D \to F(C) | C \in \mathcal{C}_0)$, and the naturality requirement in this case means that for every arrow $f : C \to C'$ in $\mathcal{C}$,

$$\begin{tikzcd} & D \arrow[dl, "\mu_C"'] \arrow[dr, "\mu_{C'}"] & \\ F(C) \arrow[rr, "F(f)"'] & & F(C') \end{tikzcd}$$

commutes in $\mathcal{D}$ (this diagram explains, I hope, the name "cone"). Let us denote the cone by (D, μ). D is called the *vertex* of the cone.

A *map of cones* $(D, \mu) \to (D', \mu')$ is a map $g : D \to D'$ such that $\mu'_C g = \mu_C$ for all $C \in \mathcal{C}_0$.

Clearly, there is a category Cone(F) which has as objects the cones for F and as morphisms maps of cones.

A *limiting cone* for F is a terminal object in Cone(F). Since terminal objects are unique up to isomorphism, as we have seen, any two limiting cones are isomorphic in Cone(F) and in particular, their vertices are isomorphic in $\mathcal{D}$.

A functor $F : \mathcal{C} \to \mathcal{D}$ is also called a *diagram* in $\mathcal{D}$ *of type* $\mathcal{C}$, and $\mathcal{C}$ is the *index category* of the diagram.

Let us see what it means to be a limiting cone, in some simple, important cases.

i) A limiting cone for the unique functor $! : \mathbf{0} \to \mathcal{D}$ ($\mathbf{0}$ is the empty category) "is" a terminal object in $\mathcal{D}$. For every object D of $\mathcal{D}$ determines, together with the empty family, a cone for !, and a map of cones is just an arrow in $\mathcal{D}$. So Cone(!) is isomorphic to $\mathcal{D}$.

ii) Let $\mathbf{2}$ be the discrete category with two objects x, y. A functor $\mathbf{2} \to \mathcal{D}$ is just a pair $\langle A, B \rangle$ of objects of $\mathcal{D}$, and a cone for this functor consists of an object C of $\mathcal{D}$ and two maps $\begin{tikzcd} C \arrow[r, "\mu_A"] \arrow[dr, "\mu_B"'] & A \\ & B \end{tikzcd}$ since there are no nontrivial arrows in $\mathbf{2}$.

$(C, (\mu_A, \mu_B))$ is a limiting cone for $\langle A, B \rangle$ iff the following holds: for any object D and arrows $f : D \to A$, $g : D \to B$, there is a unique arrow

$h : D \to C$ such that

$$\begin{tikzcd} & C \arrow[d, "\mu_A"] \arrow[ddr, "\mu_B"] & \\ D \arrow[ur, dashed, "h"] \arrow[r, "f"] \arrow[drr, "g"'] & A & \\ & & B \end{tikzcd}$$

commutes. In other words, there is, for any D, a 1-1 correspondence between maps $D \to C$ and pairs of maps

$$\begin{tikzcd} & D \arrow[dl] \arrow[dr] & \\ A & & B \end{tikzcd}$$

This is the property of a *product*; a limiting cone for $\langle A, B\rangle$ is therefore called a product cone, and usually denoted:

$$\begin{tikzcd} & A \times B \arrow[dl, "\pi_A"'] \arrow[dr, "\pi_B"] & \\ A & & B \end{tikzcd}$$

The arrows π_A and π_B are called *projections.*

iii) Let $\hat{\mathbf{2}}$ denote the category $x \overset{a}{\underset{b}{\rightrightarrows}} y$. A functor $\hat{\mathbf{2}} \to \mathcal{D}$ is the same thing as a parallel pair of arrows $A \overset{f}{\underset{g}{\rightrightarrows}} B$ in $\mathcal{D}$; I write $\langle f, g\rangle$ for this functor. A cone for $\langle f, g\rangle$ is:

$$\begin{tikzcd} & D \arrow[dl, "\mu_A"'] \arrow[dr, "\mu_B"] & \\ A \arrow[rr, shift left, "f"] \arrow[rr, shift right, "g"'] & & B \end{tikzcd}$$

But $\mu_B = f\mu_A = g\mu_A$ is already defined from μ_A, so giving a cone is the same as giving a map $\mu_A : D \to A$ such that $f\mu_A = g\mu_A$. Such a cone is limiting iff for any other map $h : C \to A$ with $fh = gh$, there is a unique $k : C \to D$ such that $h = \mu_A k$.

We call μ_A, if it is limiting, an *equalizer* of the pair f, g, and the diagram $D \xrightarrow{\mu_A} A \overset{f}{\underset{g}{\rightrightarrows}} B$ an equalizer diagram.

In Sets, an equalizer of f, g is isomorphic (as a cone) to the inclusion of $\{a \in A | f(a) = g(a)\}$ into A. In categorical interpretations of logical systems (see chapters 4 and 7), equalizers are used to interpret *equality between terms.*

Exercise 29. Show that every equalizer is a monomorphism.

Exercise 30. If $E \xrightarrow{e} X \underset{g}{\overset{f}{\rightrightarrows}} Y$ is an equalizer diagram, show that e is an isomorphism if and only if $f = g$.

Exercise 31. Show that in Set, every monomorphism fits into an equalizer diagram.

iv) Let J denote the category
$$\begin{tikzcd} & y \arrow[d, "b"] \\ x \arrow[r, "a"'] & z \end{tikzcd}$$
A functor $F : J \to \mathcal{D}$ is specified by giving two arrows in $\mathcal{D}$ with the same codomain, say $f : X \to Z$, $g : Y \to Z$. A limit for such a functor is given by an object W together with projections
$$\begin{tikzcd} W \arrow[r, "p_Y"] \arrow[d, "p_X"'] & Y \\ X & \end{tikzcd}$$
satisfying $fp_X = gp_Y$, and such that, given any other pair of arrows:
$$\begin{tikzcd} V \arrow[r, "r"] \arrow[d, "s"'] & Y \\ X & \end{tikzcd}$$
with $gr = fs$, there is a unique arrow $V \to W$ such that
$$\begin{tikzcd} V \arrow[dr] \arrow[drr, "r"] \arrow[ddr, "s"'] & & \\ & W \arrow[r, "p_Y"'] \arrow[d, "p_X"] & Y \arrow[d, "g"] \\ & X \arrow[r, "f"'] & Z \end{tikzcd}$$
commutes.

The diagram
$$\begin{tikzcd} W \arrow[r, "p_Y"] \arrow[d, "p_X"'] & Y \arrow[d, "g"] \\ X \arrow[r, "f"'] & Z \end{tikzcd}$$

is called a *pullback diagram*. In Set, the pullback cone for f, g is isomorphic to

$$\{(x, y) \in X \times Y \,|\, f(x) = g(y)\}$$

with the obvious projections.

We say that a category $\mathcal{D}$ *has binary products* (equalizers, pullbacks) iff every functor $\mathbf{2} \to \mathcal{D}$ ($\hat{\mathbf{2}} \to \mathcal{D}$, $J \to \mathcal{D}$, respectively) has a limiting cone. Some dependencies hold in this context:

Proposition 3.1 *If a category $\mathcal{D}$ has a terminal object and pullbacks, it has binary products and equalizers.*
If $\mathcal{D}$ has binary products and equalizers, it has pullbacks.

Proof. Let 1 be the terminal object in $\mathcal{D}$; given objects X and Y, if
$$\begin{tikzcd} C \arrow[r, "p_X"] \arrow[d, "p_Y"'] & X \arrow[d] \\ Y \arrow[r] & 1 \end{tikzcd}$$
is a pullback diagram, then
$$\begin{tikzcd} C \arrow[r, "p_X"] \arrow[d, "p_Y"'] & X \\ Y & \end{tikzcd}$$
is a product cone.

Given a product cone
$$\begin{tikzcd} A \times B \arrow[r, "\pi_A"] \arrow[d, "\pi_B"'] & A \\ B & \end{tikzcd}$$
and maps
$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g"'] & A \\ B & \end{tikzcd}$$
we write $X \stackrel{\langle f,g \rangle}{\to} A \times B$ for the unique factorization through the product. Write also $\delta : Y \to Y \times Y$ for $\langle \mathrm{id}_Y, \mathrm{id}_Y \rangle$.

Now given $f, g : X \to Y$, if

$$\begin{tikzcd} E \arrow[r, "e"] \arrow[d] & X \arrow[d, "\langle f,g \rangle"] \\ Y \arrow[r, "\delta"'] & Y \times Y \end{tikzcd}$$

is a pullback diagram, then $E \xrightarrow{e} X \underset{g}{\overset{f}{\rightrightarrows}} Y$ is an equalizer diagram. This proves the first statement.

As for the second: given
$$\begin{tikzcd} & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z \end{tikzcd}$$
let $E \xrightarrow{e} X \times Y \underset{g\pi_Y}{\overset{f\pi_X}{\rightrightarrows}} Z$ be an

equalizer; then

$$\begin{tikzcd} E \arrow[r, "\pi_X e"] \arrow[d, "\pi_Y e"'] & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z \end{tikzcd}$$

is a pullback diagram. ∎

Exercise 32. Let

$$\begin{tikzcd} A \arrow[r, "b"] \arrow[d, "a"'] & B \arrow[d, "f"] \\ X \arrow[r, "g"'] & Y \end{tikzcd}$$

a pullback diagram with f mono. Show that a is also mono. Also, if f is iso (an isomorphism), so is a.

Exercise 33. Given:

$$\begin{tikzcd} A \arrow[r, "b"] \arrow[d, "a"'] & B \arrow[r, "c"] \arrow[d, "f"'] & C \arrow[d, "d"'] \\ X \arrow[r, "g"'] & Y \arrow[r, "h"'] & Z \end{tikzcd}$$

a) if both squares are pullback squares, then so is the composite square

$$\begin{tikzcd} A \arrow[r, "cb"] \arrow[d, "a"'] & C \arrow[d, "d"] \\ X \arrow[r, "hg"'] & Z \end{tikzcd}$$

b) If the right hand square and the composite square are pullbacks, then so is the left hand square.

Exercise 34. $f : A \to B$ is a monomorphism if and only if

$$\begin{tikzcd} A \arrow[r, "\mathrm{id_A}"] \arrow[d, "\mathrm{id_A}"'] & A \arrow[d, "f"] \\ A \arrow[r, "f"'] & B \end{tikzcd}$$

is a pullback diagram.

A monomorphism $f : A \to B$ which fits into an equalizer diagram

$$A \xrightarrow{f} B \underset{h}{\overset{g}{\rightrightarrows}} C$$

is called a *regular mono*.

Exercise 35. If

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "b"'] & X \arrow[d, "g"] \\ B \arrow[r, "f"'] & Y \end{tikzcd}$$

is a pullback and g is regular mono, so is b.

Exercise 36. If f is regular mono and epi, f is iso. Every split mono is regular.

Exercise 37. Give an example of a category in which not every mono is regular.

Exercise 38. In Grp, every mono is regular [This is not so easy].

Exercise 39. In Pos, every mono is regular.

Exercise 40. If a category $\mathcal{D}$ has binary products and a terminal object, it has *all finite products*, i.e. limiting cones for every functor into $\mathcal{D}$ from a finite discrete category.

Exercise 41. Suppose $\mathcal{C}$ has binary products and suppose for every ordered pair (A, B) of objects of $\mathcal{C}$ a product cone $\begin{tikzcd} A \times B \arrow[r, "\pi_A"] \arrow[d, "\pi_B"'] & A \\ B & \end{tikzcd}$ has been chosen.

a) Show that there is a functor: $\mathcal{C} \times \mathcal{C} \xrightarrow{-\times-} \mathcal{C}$ (the product functor) which sends each pair (A, B) of objects to $A \times B$ and each pair of arrows $(f : A \to A', g : B \to B')$ to $f \times g = \langle f\pi_A, g\pi_B \rangle$.

b) From a), there are functors:

$$\mathcal{C} \times \mathcal{C} \times \mathcal{C} \underset{-\times(-\times-)}{\overset{(-\times-)\times-}{\rightrightarrows}} \mathcal{C}$$

sending (A,B,C) to $\begin{matrix}(A\times B)\times C\\ A\times(B\times C)\end{matrix}$ Show that there is a natural transformation $a=(a_{A,B,C}|A,B,C\in\mathcal{C}_0)$ from $(-\times-)\times-$ to $-\times(-\times-)$ such that for any four objects A,B,C,D of $\mathcal{C}$:

$$\begin{tikzcd}
((A\times B)\times C)\times D \arrow[rr, "a_{A\times B,C,D}"] \arrow[d, "a_{A,B,C}\times\mathrm{id}_D"'] & & (A\times B)\times(C\times D) \arrow[d, "a_{A,B,C\times D}"] \\
(A\times(B\times C))\times D \arrow[dr, "a_{A,B\times C,D}"'] & & A\times(B\times(C\times D)) \\
& A\times((B\times C)\times D) \arrow[ur, "\mathrm{id}_A\times a_{B,C,D}"'] &
\end{tikzcd}$$

commutes (This diagram is called "MacLane's pentagon").

A functor $F:\mathcal{C}\to\mathcal{D}$ is said to *preserve limits of type* $\mathcal{E}$ if for all functors $M:\mathcal{E}\to\mathcal{C}$, if (D,μ) is a limiting cone for M in $\mathcal{C}$, then $(FD,F\mu=(F(\mu_E)|E\in\mathcal{E}_0))$ is a limiting cone for FM in $\mathcal{D}$.

So, a functor $F:\mathcal{C}\to\mathcal{D}$ preserves binary products if for every product diagram
$\begin{tikzcd} A\times B \arrow[r, "\pi_B"] \arrow[d, "\pi_A"'] & B \\ A & \end{tikzcd}$
its F-image
$\begin{tikzcd} F(A\times B) \arrow[r, "F(\pi_B)"] \arrow[d, "F(\pi_A)"'] & F(B) \\ F(A) & \end{tikzcd}$
is again a product diagram. Similarly for equalizers and pullbacks.

Some more terminology: F is said to *preserve all finite limits* if it preserves limits of type $\mathcal{E}$ for every finite $\mathcal{E}$. A category which has all finite limits is called *lex* (*left exact*), *cartesian* or *finitely complete*.

A category is called *complete* if it has limits of type $\mathcal{E}$ for all small $\mathcal{E}$.

In general, limits over large (i.e. not small) diagrams do not exist. For example in Set, there is a limiting cone for the identity functor Set $\to$ Set (its vertex is the empty set), but not for the inclusion functor of the subcategory of all nonempty sets into Set.

Exercise 42. If a category $\mathcal{C}$ has equalizers, it has *all finite equalizers*: for every category $\mathcal{E}$ of the form

$$\begin{tikzcd} X \arrow[r, shift left=2, "f_1"] \arrow[r, phantom, "\vdots"] \arrow[r, shift right=2, "f_n"'] & Y \end{tikzcd}$$

every functor $\mathcal{E}\to\mathcal{C}$ has a limiting cone.

Exercise 43. Suppose $F : \mathcal{C} \to \mathcal{D}$ preserves equalizers (and $\mathcal{C}$ has equalizers) and reflects isomorphisms. Then F is faithful.

Exercise 44. Let $\mathcal{C}$ be a category with finite limits. Show that for every object C of $\mathcal{C}$, the slice category $\mathcal{C}/C$ (example j) of 1.1) has binary products: the vertex of a product diagram for two objects $D \to C$, $D' \to C$ is $D'' \to C$ where

$$\begin{tikzcd} D'' \arrow[r] \arrow[d] & D \arrow[d] \\ D' \arrow[r] & C \end{tikzcd}$$

is a pullback square in $\mathcal{C}$.

3.2 Limits by products and equalizers

In Set, every small diagram has a limit; given a functor $F : \mathcal{E} \to \text{Set}$ with $\mathcal{E}$ small, there is a limiting cone for F in Set with vertex

$$\{(x_E)_{E\in\mathcal{E}_0} \in \prod_{E\in\mathcal{E}_0} F(E) | \forall E \xrightarrow{f} E' \in \mathcal{E}_1 (F(f)(x_E) = x_{E'})\}$$

So in Set, limits are equationally defined subsets of suitable products. This holds in any category:

Proposition 3.2 *Suppose $\mathcal{C}$ has all small products (including the empty product, i.e. a terminal object 1) and equalizers; then $\mathcal{C}$ has all small limits.*

Proof. Given a set I and an I-indexed family of objects $(A_i | i \in I)$ of $\mathcal{C}$, we denote the product by $\prod_{i\in I} A_i$ and projections by $\pi_i : \prod_{i\in I} A_i \to A_i$; an arrow $f : X \to \prod_{i\in I} A_i$ which is determined by the compositions $f_i = \pi_i f : X \to A_i$, is also denoted $(f_i | i \in I)$.

Now given $\mathcal{E} \to \mathcal{C}$ with $\mathcal{E}_0$ and $\mathcal{E}_1$ sets, we construct

$$\begin{tikzcd} E \arrow[r, "e"] & \prod_{i\in\mathcal{E}_0} F(i) \arrow[r, shift left, "(\pi_{\mathrm{cod}(u)} | u\in\mathcal{E}_1)"] \arrow[r, shift right, "(F(u)\pi_{\mathrm{dom}(u)} | u\in\mathcal{E}_1)"'] & \prod_{u\in\mathcal{E}_1} F(\mathrm{cod}(u)) \end{tikzcd}$$

in $\mathcal{C}$ as an equalizer diagram. The family $(\mu_i = \pi_i e : E \to F(i) | i \in \mathcal{E}_0)$ is a natural transformation $\Delta_E \Rightarrow F$ because, given an arrow $u \in \mathcal{E}_1$, say $u : i \to j$, we have that

$$\begin{tikzcd} & E \arrow[dl, "\pi_i e"'] \arrow[dr, "\pi_j e"] & \\ F(i) \arrow[rr, "F(u)"'] & & F(j) \end{tikzcd}$$

commutes since $F(u)\pi_i e = F(u)\pi_{\mathrm{dom}(u)}e = \pi_{\mathrm{cod}(u)}e = \pi_j e$.

So (E, μ) is a cone for F, but every other cone (D, ν) for F gives a map $d : D \to \prod_{i \in \mathcal{E}_0} F(i)$ equalizing the two horizontal arrows; so factors uniquely through E. ∎

Exercise 45. Check that "small" in the statement of the proposition, can be replaced by "finite": if $\mathcal{C}$ has all finite products and equalizers, $\mathcal{C}$ is finitely complete.

3.3 Colimits

The dual notion of limit is colimit. Given a functor $F : \mathcal{E} \to \mathcal{C}$ there is clearly a functor $F^{\mathrm{op}} : \mathcal{E}^{\mathrm{op}} \to \mathcal{C}^{\mathrm{op}}$ which does "the same" as F. We say that a *colimiting cocone* for F is a limiting cone for F^{op}.

So: a cocone for $F : \mathcal{E} \to \mathcal{C}$ is a pair (ν, D) where $\nu : F \Rightarrow \Delta_D$ and a colimiting cocone is an initial object in the category Cocone(F).

Examples

i) a colimiting cocone for $! : \mathbf{0} \to \mathcal{C}$ "is" an initial object of $\mathcal{C}$

ii) a colimiting cocone for $\langle A, B \rangle : \mathbf{2} \to \mathcal{C}$ is a *coproduct* of A and B in $\mathcal{C}$: usually denoted $A + B$ or $A \sqcup B$; there are *coprojections* or *coproduct inclusions*

$$\begin{tikzcd} A \arrow[dr, "\nu_A"] & \\ B \arrow[r, "\nu_B"'] & A \sqcup B \end{tikzcd}$$

with the property that, given any pair of arrows $A \xrightarrow{f} C$, $B \xrightarrow{g} C$ there is a unique map $\begin{bmatrix} f \\ g \end{bmatrix} : A \sqcup B \to C$ such that $f = \begin{bmatrix} f \\ g \end{bmatrix} \nu_A$ and $g = \begin{bmatrix} f \\ g \end{bmatrix} \nu_B$

iii) a colimiting cocone for $\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \end{tikzcd}$ (as functor $\hat{\mathbf{2}} \to \mathcal{C}$) is given by a map $B \xrightarrow{c} C$ satisfying $cf = cg$, and such that for any $B \xrightarrow{h} D$ with $hf = hg$ there is a unique $C \xrightarrow{h'} D$ with $h = h'c$. c is called a *coequalizer* for f and g; the diagram $\begin{tikzcd} A \arrow[r, shift left] \arrow[r, shift right] & B \arrow[r] & C \end{tikzcd}$ a coequalizer diagram.

Exercise 46. Is the terminology "coproduct inclusions" correct? That is, it suggests they are monos. Is this always the case?

In Set, the coproduct of X and Y is the disjoint union $(\{0\} \times X) \cup (\{1\} \times Y)$

of X and Y. The coequalizer of $X \overset{f}{\underset{g}{\rightrightarrows}} Y$ is the quotient map $Y \to Y/\sim$ where $\sim$ is the equivalence relation generated by

$$y \sim y' \text{ iff there is } x \in X \text{ with } f(x) = y \text{ and } g(x) = y'$$

The dual notion of pullback is *pushout*. A pushout diagram is a colimiting cocone for a functor $\Gamma \to \mathcal{C}$ where Γ is the category

$$\begin{tikzcd} x \arrow[r] \arrow[d] & y \\ z & \end{tikzcd}$$

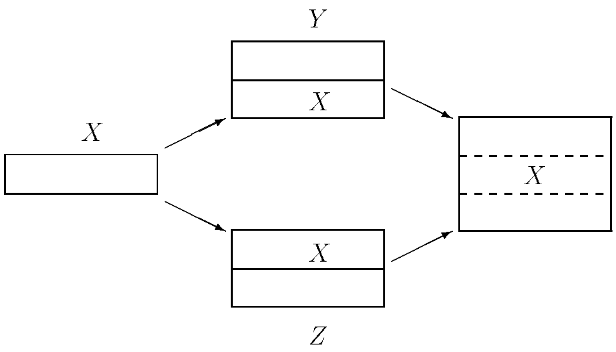
. Such a diagram is a square

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g"'] & Y \arrow[d, "a"] \\ Z \arrow[r, "b"'] & P \end{tikzcd}$$

which commutes and such that, given

$$\begin{tikzcd} & Y \arrow[d, "\alpha"] \\ Z \arrow[r, "\beta"'] & Q \end{tikzcd}$$

with $\alpha f = \beta g$, there is a unique $P \xrightarrow{p} Q$ with $\alpha = pa$ and $\beta = pb$. In Set, the pushout of $X \xrightarrow{f} Y$ and $X \xrightarrow{g} Z$ is the coproduct $Y \sqcup Z$ where the two images of X are identified:



Exercise 47. Give yourself, in terms of $X \xrightarrow{f} Y$ and $X \xrightarrow{g} Z$, a formal definition of a relation R on $Y \sqcup Z$ such that the pushout of f and g is $Y \sqcup Z/\sim$, $\sim$ being the equivalence relation generated by R.

One can now dualize every result and exercise from the section on limits:

Exercise 48. f is epi if and only if

$$\begin{tikzcd} \cdot \arrow[r, "f"] \arrow[d, "f"'] & \cdot \arrow[d, "\mathrm{id}"] \\ \cdot \arrow[r, "\mathrm{id}"'] & \cdot \end{tikzcd}$$

is a pushout diagram.

Exercise 49. Every coequalizer is an epimorphism; if e is a coequalizer of f and g, then e is iso iff $f = g$

Exercise 50. If $\mathcal{C}$ has an initial object and pushouts, $\mathcal{C}$ has binary coproducts and coequalizers; if $\mathcal{C}$ has binary coproducts and coequalizers, $\mathcal{C}$ has pushouts.

Exercise 51. If $\begin{tikzcd} \cdot \arrow[r] \arrow[d, "a"'] & \cdot \arrow[d, "f"] \\ \cdot \arrow[r] & \cdot \end{tikzcd}$ is a pushout diagram, then a epi implies f epi, and a regular epi (i.e. a coequalizer) implies f regular epi.

Exercise 52. The composition of two puhout squares is a pushout; if both the first square and the composition are pushouts, the second square is.

Exercise 53. If $\mathcal{C}$ has all small (finite) coproducts and coequalizers, $\mathcal{C}$ has all small (finite) colimits.

Exercise 54. In Pos, $X \xrightarrow{f} Y$ is a regular epi if and only if for all y, y' in Y:

$$y \leq y' \Leftrightarrow \exists x \in f^{-1}(y) \exists x' \in f^{-1}(y').x \leq x'$$

Show by an example that not every epi is regular in Pos.

Exercise 55. In Grp, every epi is regular.

4 A little piece of categorical logic

One of the major achievements of category theory in mathematical logic and in computer science, has been a unified treatment of semantics for all kinds of logical systems and term calculi which are the basis for programming languages.

One can say that mathematical logic, seen as the study of classical first order logic, first started to be a real subject with the discovery, by Gödel, of the *completeness theorem* for set-theoretic interpretations: a sentence φ is provable if and only if φ is true in all possible interpretations. This unites the two approaches to logic: proof theory and model theory, makes logic accessible for mathematical methods and enables one to give nice and elegant proofs of proof theoretical properties by model theory (for example, the Beth and Craig definability and interpolation theorems).

However the completeness theorem needs generalization once one considers logics, such as intuitionistic logic (which does not admit the principle of excluded middle), minimal logic (which has no negation) or modal logic (where the logic has an extra operator, expressing "necessarily true"), for which the set-theoretic interpretation is not complete. One therefore comes with a general definition of "interpretation" in a category $\mathcal{C}$ of a logical system, which generalizes Tarski's truth definition: this will then be the special case of classical logic and the category Set.

In this chapter I treat, for reasons of space, only a fragment of first order logic: coherent logic. On this fragment the valid statements of classical and intuitionistic logic coincide.

For an interpretation of a term calculus like the λ-calculus, which is of paramount importance in theoretical computer science, the reader is referred to chapter 7.

4.1 Regular categories and subobjects

Definition 4.1 *A category $\mathcal{C}$ is called* regular *if the following conditions hold:*

a) $\mathcal{C}$ has all finite limits;

b) For every arrow f, if

$$\begin{tikzcd} Z \arrow[r, "p_0"] \arrow[d, "p_1"'] & X \arrow[d, "f"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

is a pullback (then $Z \overset{p_0}{\underset{p_1}{\rightrightarrows}} X$ *is called the* kernel pair *of f), the coequalizer of p_0, p_1 exists;*

c) Regular epimorphisms (coequalizers) are stable under pullback, *that is: in a pullback square*

$$\begin{tikzcd} \cdot \arrow[r] \arrow[d, "a"'] & \cdot \arrow[d, "f"] \\ \cdot \arrow[r] & \cdot \end{tikzcd}$$

if f is regular epi, so is a.

Exercise 56. Set is regular. Prove that Pos is regular. Show that Top is not regular [the case of Top may cause you some trouble; don't worry].

Proposition 4.2 *In a regular category, every arrow $f : X \to Y$ can be factored as $f = me : X \xrightarrow{e} E \xrightarrow{m} Y$ where e is regular epi and m is mono; and this factorization is unique in the sense that if f is also $m'e' : X \xrightarrow{e'} E' \xrightarrow{m'} Y$ with m' mono and e' regular epi, there is an isomorphism $\sigma : E \to E'$ such that $\sigma e = e'$ and $m'\sigma = m$.*

Proof. Given $f : X \to Y$ we let $X \xrightarrow{e} E$ be the coequalizer of the kernel pair $Z \underset{p_1}{\overset{p_0}{\rightrightarrows}} X$ of f. Since $fp_0 = fp_1$ there is a unique $m : E \to Y$ such that $f = me$. By construction e is regular epi; we must show that m is mono, and the uniqueness of the factorization.

Suppose $mg = mh$ for $g, h : W \to E$; we prove that $g = h$. Let

$$\begin{tikzcd} V \arrow[r, "a"] \arrow[d, "\langle q_0,q_1\rangle"'] & W \arrow[d, "\langle g,h\rangle"] \\ X \times X \arrow[r, "e\times e"'] & E \times E \end{tikzcd}$$

be a pullback square. Then

$$fq_0 = meq_0 = mga = mha = meq_1 = fq_1$$

so there is a unique arrow $V \xrightarrow{b} Z$ such that $\langle q_0, q_1\rangle = \langle p_0, p_1\rangle b : V \to X \times X$ (because of the kernel pair property). It follows that

$$ga = eq_0 = ep_0b = ep_1b = eq_1 = ha$$

I claim that a is epi, so it follows that $g = h$. It is here that we use the requirement that regular epis are stable under pullback. Now $e \times e : X \times X \to E \times E$ is the composite

$$X \times X \xrightarrow{e\times \mathrm{id}_X} E \times X \xrightarrow{\mathrm{id}_E \times e} E \times E$$

and both maps are regular epis since both squares

$$\begin{tikzcd} X \times X \arrow[r, "e\times \mathrm{id}_X"] \arrow[d, "\pi_0"'] & E \times X \arrow[d, "\pi_0"] \\ X \arrow[r, "e"'] & E \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} E \times X \arrow[r, "\mathrm{id}_E\times e"] \arrow[d, "\pi_1"'] & E \times E \arrow[d, "\pi_1"] \\ X \arrow[r, "e"'] & E \end{tikzcd}$$

are pullbacks. The map a, being the pullback of a composite of regular epis, is then itself the composite of regular epis (check this!), so in particular epi.

This proves that m is mono, and we have our factorization.

As to uniqueness, suppose we had another factorization $f = m'e'$ with m' mono and e' regular epi. Then $m'e'p_0 = fp_0 = fp_1 = m'e'p_1$ so since m' mono, $e'p_0 = e'p_1$. Because e is the coequalizer of p_0 and p_1, there is a unique σ: $\begin{tikzcd} {} \arrow[r, "e"] \arrow[dr, "e'"'] & {} \arrow[d, "\sigma"] \\ & {} \end{tikzcd}$ Then $m'\sigma e = m'e' = f = me$ so since e epi, $m = m'\sigma$.

Now $e' : X \to E'$ is a coequalizer; say $U \overset{k}{\underset{l}{\rightrightarrows}} X \xrightarrow{e'} E'$ is a coequalizer diagram. Then it follows that $ek = el$ (since $mek = m'e'k = m'e'l = mel$ and m mono) so there is a unique τ: $\begin{tikzcd} {} \arrow[r, "e"] \arrow[dr, "e'"'] & {} \\ & {} \arrow[u, "\tau"'] \end{tikzcd}$ Then $m\tau\sigma e = m\tau e' = me$; since m mono and e epi, $\tau\sigma = \mathrm{id}_E$. Similarly, $\sigma\tau = \mathrm{id}_{E'}$. So σ is the required isomorphism. ∎

Exercise 57. Check this detail: in a regular category $\mathcal{C}$, if $\begin{tikzcd} {} \arrow[r, "a"] \arrow[d] & {} \arrow[d] \\ {} \arrow[r, "b"'] & {} \end{tikzcd}$ is a pullback diagram and $b = c_1c_2$ with c_1 and c_2 regular epis, then $a = a_1a_2$ for some regular epis a_1, a_2.

Subobjects. In any category $\mathcal{C}$ we define a *subobject* of an object X to be an equivalence class of monomorphisms $Y \xrightarrow{m} X$, where $Y \xrightarrow{m} X$ is equivalent to $Y' \xrightarrow{m'} X$ if there is an isomorphism $\sigma : Y \to Y'$ with $m'\sigma = m$ (then $m\sigma^{-1} = m'$ follws). We say that $Y \xrightarrow{m} X$ represents a *smaller* subobject than $Y' \xrightarrow{m'} X$ if there is $\sigma : Y \to Y'$ with $m'\sigma = m$ (σ not necessarily iso; but check that σ is always mono).

The notion of subobject is the categorical version of the notion of subset in set theory. In Set, two injective functions represent the same subobject iff their images are the same; one can therefore identify subobjects with subsets. Note however, that in Set we have a "canonical" *choice* of representative for each subobject: the inclusion of the subset to which the subobject corresponds. This choice is not always available in general categories.

We have a partial order Sub(X) of subobjects of X, ordered by the smaller-than relation.

Proposition 4.3 *In a category $\mathcal{C}$ with finite limits, each pair of elements of Sub(X) has a greatest lower bound. Moreover, Sub(X) has a largest element.*

Proof. If $Y \xrightarrow{m} X$ and $Y' \xrightarrow{m'} X$ represent two subobjects of X and
$$\begin{tikzcd} Z \arrow[r] \arrow[d] & Y \arrow[d, "m"] \\ Y' \arrow[r, "m'"'] & X \end{tikzcd}$$
is a pullback, then $Z \to X$ is mono, and represents the greatest lower bound (check!).

Of course, the identity $X \xrightarrow{\mathrm{id}_X} X$ represents the top element of Sub(X). ∎

Because the factorization of $X \xrightarrow{f} Y$ as $X \xrightarrow{e} E \xrightarrow{m} Y$ with e regular epi and m mono, in a regular category $\mathcal{C}$, is only defined up to isomorphism, it defines rather a subobject of Y than a mono into Y; this defined subobject is called the *image* of f and denoted Im(f) (compare with the situation in Set).

Exercise 58. Im(f) is the smallest subobject of Y through which f factors: for a subobject represented by $n : A \to Y$ we have that there is $X \xrightarrow{a} A$ with $f = na$, if and only if Im(f) is smaller than the subobject represented by n.

Since monos and isos are stable under pullback, in any category $\mathcal{C}$ with pullbacks, any arrow $f : X \to Y$ determines an order preserving map $f^* : \mathrm{Sub}(Y) \to \mathrm{Sub}(X)$ *by pullback along* f: if $E \xrightarrow{m} Y$ represents the subobject M of Y and
$$\begin{tikzcd} F \arrow[r] \arrow[d, "n"'] & E \arrow[d, "m"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$
is a pullback, $F \xrightarrow{n} X$ represents $f^*(M)$.

Exercise 59. Check that f^* is well defined and order preserving.

Proposition 4.4 *In a regular category, each f^* preserves greatest lower bounds and images, that is: for $f : X \to Y$,*

i) for subobjects M, N of Y, $f^*(M \wedge N) = f^*(M) \wedge f^*(N)$;

ii) if
$$\begin{tikzcd} \cdot \arrow[r] \arrow[d, "g'"'] & \cdot \arrow[d, "g"] \\ \cdot \arrow[r, "f"'] & \cdot \end{tikzcd}$$
is a pullback, then $f^*(\mathrm{Im}(g)) = \mathrm{Im}(g')$.

Exercise 60. Prove proposition 4.4.

4.2 Coherent logic in regular categories

The fragment of first order logic we are going to interpret in regular categories is the so-called coherent fragment.

The logical symbols are $=$ (equality), $\wedge$ (conjunction) and $\exists$ (existential quantification). A *language* consists of a set of *sorts* $S, T, \ldots$; a denumerable collection of *variables* $x_1^S, x_2^S, \ldots$ of sort S, for each sort; and a collection of *function symbols* $(f : S_1, \ldots, S_n \to S)$. The case $n = 0$ is not excluded (one thinks of *constants* of a sort), but not separately treated. We now define, inductively, terms of sort S and formulas.

Definition 4.5 *Terms of sort S are defined by:*

i) x^S *is a term of sort* S *if* x^S *is a variable of sort* S*;*

ii) *if* $t_1, \ldots, t_n$ *are terms of sorts* $S_1, \ldots, S_n$ *respectively, and*

$$(f : S_1, \ldots S_n \to S)$$

is a function symbol of the language, then $f(t_1, \ldots, t_n)$ *is a term of sort* S*.*

Formulas are defined by:

i) $\top$ *is a formula (the formula "true");*

ii) *if* t *and* s *are terms of the same sort, then* $t = s$ *is a formula;*

iii) *if* φ *and* ψ *are formulas then* $(\varphi \wedge \psi)$ *is a formula;*

iv) *if* φ *is a formula and* x *a variable of some sort, then* $\exists x \varphi$ *is a formula.*

An *interpretation* of such a language in a regular category $\mathcal{C}$ starts by choosing for each sort S an object $[\![S]\!]$ of $\mathcal{C}$, and for each function symbol $(f : S_1, \ldots, S_n \to S)$ of the language, an arrow $[\![f]\!] : [\![S_1]\!] \times \cdots \times [\![S_n]\!] \to [\![S]\!]$ in $\mathcal{C}$.

Given this, we define interpretations $[\![t]\!]$ for terms t and $[\![\varphi]\!]$ for formulas φ, as follows.

Write $FV(t)$ for the set of variables which occur in t, and $FV(\varphi)$ for the set of *free* variables in φ.

We put $[\![FV(t)]\!] = [\![S_1]\!] \times \cdots \times [\![S_n]\!]$ if $FV(t) = \{x_1^{S_1}, \ldots, x_n^{S_n}\}$; the same for $[\![FV(\varphi)]\!]$. Note: in the products $[\![FV(t)]\!]$ and $[\![FV(\varphi)]\!]$ we take a copy of $[\![S]\!]$ for every variable of sort S! Let me further emphasize that the empty product is 1, so if $FV(t)$ (or $FV(\varphi)$) is $\emptyset$, $[\![FV(t)]\!]$ (or $[\![FV(\varphi)]\!]$) is the terminal object of the category.

Definition 4.6 *The interpretation of a term t of sort S is a morphism $[\![t]\!] : [\![FV(t)]\!] \to [\![S]\!]$ and is defined by the clauses:*

i) $[\![x^S]\!]$ *is the identity on* $[\![S]\!]$*, if* x^S *is a variable of sort* S*;*

ii) *Given* $[\![t_i]\!] : [\![FV(t_i)]\!] \to [\![S_i]\!]$ *for* $i = 1, \ldots, n$ *and a function symbol* $(f : S_1, \ldots, S_n \to S)$ *of the language,* $[\![f(t_1, \ldots, t_n)]\!]$ *is the map*

$$[\![FV(f(t_1,\ldots,t_n))]\!] \xrightarrow{(\tilde{t}_i|i=1,\ldots,n)} \prod_{i=1}^n [\![S_i]\!] \xrightarrow{[\![f]\!]} [\![S]\!]$$

where $\tilde{t}_i$ *is the composite*

$$[\![FV(f(t_1,\ldots,t_n))]\!] \xrightarrow{\pi_i} [\![FV(t_i)]\!] \xrightarrow{[\![t_i]\!]} [\![S_i]\!]$$

in which π_i *is the appropriate projection, corresponding to the inclusion* $FV(t_i) \subseteq FV(f(t_1, \ldots, t_n))$*.*

Finally, we interpret formulas φ as *subobjects* $[\![\varphi]\!]$ of $[\![FV(\varphi)]\!]$. You should try to keep in mind the intuition that $[\![\varphi(x_1, \ldots, x_n)]\!]$ is the "subset"

$$\{(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n | \varphi[a_1, \ldots, a_n]\}$$

Definition 4.7 *The interpretation* $[\![\varphi]\!]$ *as subobject of* $[\![FV(\varphi)]\!]$ *is defined as follows:*

i) $[\![\top]\!]$ *is the maximal subobject of* $[\![FV(\top)]\!] = 1$*;*

ii) $[\![t = s]\!] \to [\![FV(t = s)]\!]$ *is the equalizer of*

$$[\![FV(t=s)]\!] \rightrightarrows \begin{matrix}[\![FV(t)]\!] \xrightarrow{[\![t]\!]} \\ [\![FV(s)]\!] \xrightarrow[{[\![s]\!]}]{} \end{matrix} [\![T]\!]$$

if t *and* s *are of sort* T*; again, the left hand side maps are projections, corresponding to the inclusions of* $FV(t)$ *and* $FV(s)$ *into* $FV(t = s)$*;*

iii) *if* $[\![\varphi]\!] \to [\![FV(\varphi)]\!]$ *and* $[\![\psi]\!] \to [\![FV(\psi)]\!]$ *are given and*

$$\begin{tikzcd}
{[\![FV(\varphi\wedge\psi)]\!]} \arrow[r, "\pi_1"] \arrow[dr, "\pi_2"'] & {[\![FV(\varphi)]\!]} \\
& {[\![FV(\psi)]\!]}
\end{tikzcd}$$

are again the suitable projections, then $[\![(\varphi \wedge \psi)]\!] \to [\![FV(\varphi \wedge \psi)]\!]$ *is the greatest lower bound in* $\mathrm{Sub}([\![FV(\varphi \wedge \psi)]\!])$ *of* $\pi_1^*([\![\varphi]\!])$ *and* $\pi_2^*([\![\psi]\!])$*;*

iv) *if* $[\![\varphi]\!] \to [\![FV(\varphi)]\!]$ *is given and* $\pi : [\![FV(\varphi)]\!] \to [\![FV(\exists x\varphi)]\!]$ *the projection, let* $[\![FV'(\varphi)]\!]$ *be the product of the interpretations of the sorts of the variables in* $FV(\varphi) \cup \{x\}$ *(so* $[\![FV'(\varphi)]\!] = [\![FV(\varphi)]\!]$ *if* x *occurs freely in* φ*; and* $[\![FV'(\varphi)]\!] = [\![FV(\varphi)]\!] \times [\![S]\!]$ *if* $x = x^S$ *does not occur free in* φ*). Write* $\pi' : [\![FV'(\varphi)]\!] \to [\![FV(\varphi)]\!]$.

Now take $[\![\exists x\varphi]\!] \to [\![FV(\exists x\varphi)]\!]$ *to be the image of the composition:*

$$(\pi')^*([\![\varphi]\!]) \to [\![FV'(\varphi)]\!] \stackrel{\pi\pi'}{\to} [\![FV(\exists x\varphi)]\!]$$

We have now given an interpretation of formulas. Basically, a formula φ is *true* under this interpretation if $[\![\varphi]\!] \to [\![FV(\varphi)]\!]$ is the maximal subobject; but since we formulate the logic in terms of sequents we rather define when a sequent is true under the interpretation.

Definition 4.8 *A* labelled sequent *is an expression of the form* $\psi \vdash_\sigma \varphi$ *or* $\vdash_\sigma \varphi$ *where* ψ *and* φ *are the formulas of the sequent (but* ψ *may be absent), and* σ *is a finite set of variables which includes all the variables which occur free in a formula of the sequent.*

Let $[\![\sigma]\!] = [\![S_1]\!] \times \cdots \times [\![S_n]\!]$ *if* $\sigma = \{x_1^{S_1}, \ldots, x_n^{S_n}\}$*; there are projections* $[\![\sigma]\!] \stackrel{\pi_\varphi}{\to} [\![FV(\varphi)]\!]$ *and (in case* ψ *is there)* $[\![\sigma]\!] \stackrel{\pi_\psi}{\to} [\![FV(\psi)]\!]$*; we say that the sequent* $\psi \vdash_\sigma \varphi$ *is true for the interpretation if* $(\pi_\psi)^*([\![\psi]\!]) \leq (\pi_\varphi)^*([\![\varphi]\!])$ *as subobjects of* $[\![\sigma]\!]$*, and* $\vdash_\sigma \varphi$ *is true if* $(\pi_\varphi)^*([\![\varphi]\!])$ *is the maximal subobject of* $[\![\sigma]\!]$.

Exercise 61. Show that the sequent $\vdash \exists x^S(x^S = x^S)$ is true if and only if the unique map $[\![S]\!] \to 1$ is a regular epimorphism.

We now turn to the logic. Instead of giving deduction rules and axioms, I formulate a list of closure conditions which determine what sets of labelled sequents will be called a *theory*. I write $\vdash_x$ for $\vdash_{\{x\}}$ and $\vdash$ for $\vdash_\emptyset$.

Definition 4.9 *Given a language, a set* T *of labelled sequents of that language is called a* theory *iff the following conditions hold (the use of brackets around* ψ *caters in a, I hope, self-explanatory way for the case distiction as to whether* ψ *is or is not present):*

i) $\vdash \top$ *is in* T*;*
$\vdash_x x = x$ *is in* T *for every variable* x*;*
$x = y \wedge y = z \vdash_{\{x,y,z\}} x = z$ *is in* T *for variables* x, y, z *of the same sort;*

ii) *if* $(\psi) \vdash_\sigma \varphi$ *is in* T *then* $(\psi) \vdash_\tau \varphi$ *is in* T *whenever* $\sigma \subseteq \tau$*;*

iii) *if* $(\psi) \vdash_\sigma \varphi$ *is in* T *and* $FV(\chi) \subseteq \sigma$ *then* $(\psi\wedge)\chi \vdash_\sigma \varphi$ *and* $\chi(\wedge\psi) \vdash_\sigma \varphi$ *are in* T*;*

iv) *if* $(\psi) \vdash_\sigma \varphi$ *and* $(\psi) \vdash_\sigma \chi$ *are in* T *then* $(\psi) \vdash_\sigma \varphi \wedge \chi$ *and* $(\psi) \vdash_\sigma \chi \wedge \varphi$ *are in* T*;*

v) *if* $\psi \vdash_\sigma \varphi$ *is in* T *and* x *is a variable not occurring in* φ *then* $\exists x\psi \vdash_{\sigma \setminus \{x\}} \varphi$ *is in* T*;*

vi) *if* x *occurs in* φ *and* $(\psi) \vdash_\sigma \varphi[t/x]$ *is in* T *then* $(\psi) \vdash_\sigma \exists x\varphi$ *is in* T*;*
if x *does not occur in* φ *and* $(\psi) \vdash_\sigma \varphi$ *and* $(\psi) \vdash_\sigma \exists x(x = x)$ *are in* T*, then* $(\psi) \vdash_\sigma \exists x\varphi$ *is in* T*;*

vii) *if* $(\psi) \vdash_\sigma \varphi$ *is in* T *then* $(\psi[t/x]) \vdash_{\sigma \setminus \{x\} \cup FV(t)} \varphi[t/x]$ *is in* T*;*

viii) *if* $(\psi) \vdash_\sigma \varphi[t/x]$ *and* $(\psi) \vdash_\sigma t = s$ *are in* T *then* $(\psi) \vdash_\sigma \varphi[s/x]$ *is in* T*;*

ix) *if* $(\psi) \vdash_\sigma \varphi$ *and* $\varphi \vdash \chi$ *are in* T *then* $(\psi) \vdash_\sigma \chi$ *is in* T

Exercise 62. Show that the sequent $\varphi \vdash_{FV(\varphi)} \varphi$ is in every theory, for every formula φ of the language.

As said, the definition of a theory is a list of closure conditions: every item can be seen as a rule, and a theory is a set of sequents closed under every rule. Therefore, the intersection of any collection of theories is again a theory, and it makes sense to speak, given a set of sequents S, of the theory $Cn(S)$ *generated by* S:

$$Cn(S) = \bigcap\{T | T \text{ is a theory and } S \subseteq T\}$$

We have the following theorem:

Theorem 4.10 (Soundness theorem) *Suppose* $T = Cn(S)$ *and all sequents of* S *are true under the interpretation in the category* $\mathcal{C}$*. Then all sequents of* T *are true under that interpretation.*

Before embarking on the proof, first a lemma:

Lemma 4.11 *Suppose* t *is substitutable for* x *in* φ*. There is an obvious map*

$$[t] : [\![FV(\varphi) \setminus \{x\} \cup FV(t)]\!] = [\![FV(\varphi[t/x])]\!] \to [\![FV(\varphi)]\!]$$

induced by $[\![t]\!]$*; the components of* $[t]$ *are projections except for the factor of* $[\![\varphi]\!]$ *corresponding to* x*, where it is*

$$[\![FV(\varphi[t/x])]\!] \to [\![FV(t)]\!] \stackrel{[\![t]\!]}{\to} [\![\{x\}]\!]$$

There is a pullback diagram:

$$\begin{tikzcd}
[\![\varphi[t/x]]\!] \arrow[r] \arrow[d] & [\![FV(\varphi[t/x])]\!] \arrow[d, "{[t]}"] \\
[\![\varphi]\!] \arrow[r] & [\![FV(\varphi)]\!]
\end{tikzcd}$$

Exercise 63. Prove this lemma [not trivial. Use induction on φ and proposition 4.4].

Proof. (of theorem 4.10) The proof checks that for every rule in the list of definition 4.9, if the premiss is true then the conclusion is true; in other words, that the set of true sequents is a theory.
i) $\vdash \top$ is true by the definition $\llbracket \top \rrbracket = 1$;
$\llbracket x^S = x^S \rrbracket$ is the equalizer of two maps which are both the identity on $\llbracket S \rrbracket$, so isomorphic to $\llbracket S \rrbracket$. For $x = y \wedge y = z \vdash_{\{x,y,z\}} x = z$, just observe that $E_{12} \wedge E_{23} \leq E_{13}$ if E_{ij} is the equalizer of the two projections $\pi_i, \pi_j : \llbracket S \rrbracket \times \llbracket S \rrbracket \times \llbracket S \rrbracket \to \llbracket S \rrbracket$.
ii) This is because if $\sigma \subseteq \tau$ and $\rho : \llbracket \tau \rrbracket \to \llbracket \sigma \rrbracket$ is the projection, ρ^* is monotone.
iii)-iv) By the interpretation of $\wedge$ as the greatest lower bound of two subobjects, and proposition 4.4.
v) Let

$$\begin{tikzcd}
\llbracket \sigma \rrbracket \arrow[r, "\pi"] \arrow[d, "\mu"'] & \llbracket \sigma \setminus \{x\} \rrbracket \arrow[r, "\rho"] \arrow[d, "\nu"] & \llbracket FV(\varphi) \rrbracket \\
\llbracket FV(\psi) \rrbracket & \llbracket FV(\exists x \psi) \rrbracket &
\end{tikzcd}$$

the projections. Since by assumption $\mu^*(\llbracket \psi \rrbracket) \leq (\rho\pi)^*(\llbracket \varphi \rrbracket)$ there is a commutative diagram

$$\begin{tikzcd}
\mu^*(\llbracket \psi \rrbracket) \arrow[r] \arrow[d] & \llbracket \sigma \rrbracket \arrow[d, "\pi"] \\
\rho^*(\llbracket \varphi \rrbracket) \arrow[r] & \llbracket \sigma \setminus \{x\} \rrbracket
\end{tikzcd}$$

By proposition 4.4, $\nu^*(\llbracket \exists x \psi \rrbracket)$ is the image of the map $\mu^*(\llbracket \psi \rrbracket) \to \llbracket \sigma \setminus \{x\} \rrbracket$, so $\nu^*(\llbracket \psi \rrbracket) \leq \rho^*(\llbracket \varphi \rrbracket)$ in $\mathrm{Sub}(\llbracket \sigma \setminus \{x\} \rrbracket)$.
vi) Suppose x occurs free in φ. Consider the commutative diagram

$$\begin{tikzcd}
& \llbracket \sigma \rrbracket \arrow[dl, "\pi"'] \arrow[d, "\pi'"] \arrow[drrr, "\pi''"] & & & \\
\llbracket FV(\psi) \rrbracket & \llbracket FV(\varphi[t/x]) \rrbracket \arrow[rr, "{[t]}"] & & \llbracket FV(\varphi) \rrbracket \arrow[r, "\rho"] & \llbracket FV(\varphi) \setminus \{x\} \rrbracket
\end{tikzcd}$$

with $[t]$ as in lemma 4.11 and the other maps projections. Now $\llbracket \varphi \rrbracket \leq \rho^*(\llbracket \exists x \varphi \rrbracket)$ because $\llbracket \varphi \rrbracket \to \llbracket FV(\varphi) \rrbracket \xrightarrow{\rho} \llbracket FV(\varphi) \setminus \{x\} \rrbracket$ factors through $\llbracket \exists x \varphi \rrbracket$ by definition; so if $\pi^*(\llbracket \psi \rrbracket) \leq \pi'^*(\llbracket \varphi[t/x] \rrbracket)$ then with lemma 4.11,

$$\pi^*(\llbracket \psi \rrbracket) \leq \pi'^*(\llbracket \varphi[t/x] \rrbracket) = \pi'^*[t]^*(\llbracket \varphi \rrbracket) \leq \pi'^*[t]^*\rho^*(\llbracket \exists x \varphi \rrbracket) = \pi''^*(\llbracket \exists x \varphi \rrbracket)$$

in $\mathrm{Sub}(\llbracket \sigma \rrbracket)$ and we are done.

The case of x not occurring in φ is left to you.
vii) Direct application of lemma 4.11
viii-ix) Left to you. ∎

Exercise 64. Fill in the "left to you" gaps in the proof.

4.3 The language $\mathcal{L}(\mathcal{C})$ and theory $T(\mathcal{C})$ associated to a regular category $\mathcal{C}$

Given a regular category $\mathcal{C}$ (which, to be precise, must be assumed to be small), we associate to $\mathcal{C}$ the language which has a sort C for every object of $\mathcal{C}$, and a function symbol $(f : C \to D)$ for every arrow $f : C \to D$ of $\mathcal{C}$.

This language is called $\mathcal{L}(\mathcal{C})$ and it has trivially an interpretation in $\mathcal{C}$.

The theory $T(\mathcal{C})$ is the set of sequents of $\mathcal{L}(\mathcal{C})$ which are true for this interpretation.

One of the points of categorical logic is now, that categorical statements about objects and arrows in $\mathcal{C}$ can be reformulated as statements about the truth of certain sequents in $\mathcal{L}(\mathcal{C})$. You should read the relevant sequents as expressing that we can "do as if the category were Set".

Examples

a) C is a terminal object of $\mathcal{C}$ if and only if the sequents $\vdash_{x,y} x = y$ and $\vdash \exists x(x = x)$ are valid, where x, y variables of sort C;

b) the arrow $f : A \to B$ is mono in $\mathcal{C}$ if and only if the sequent $f(x) = f(y) \vdash_{x,y} x = y$ is true;

c) The square

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \arrow[d, "h"] \\ C \arrow[r, "d"'] & D \end{tikzcd}$$

is a pullback square in $\mathcal{C}$ if and only if the sequents

$$h(x^B) = d(y^C) \vdash_{x,y} \exists z^A(f(z) = x \wedge g(z) = y)$$

and

$$f(z^A) = f(z'^A) \wedge g(z^A) = g(z'^A) \vdash_{z,z'} z = z'$$

are true;

d) the fact that $f : A \to B$ is epi can not similarly be expressed! But: f is regular epi if and only if

$$\vdash_{x^B} \exists y^A (f(y) = x)$$

is true;

e) $A \xrightarrow{f} B \overset{g}{\underset{h}{\rightrightarrows}} C$ is an equalizer diagram iff f is mono (see b) and the sequent

$$g(x^B) = h(x^B) \vdash_{x^B} \exists y^A (f(y) = x)$$

is true.

Exercise 65. Check (a number of) these statements. Give the sequent(s) corresponding to the statement that
$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "g"'] & B \\ C & \end{tikzcd}$$
is a product diagram.

Exercise 66. Check that the formulas $\exists x \varphi$ and $\exists x(x = x \wedge \varphi)$ are *equivalent*, that is, every theory contains the sequents

$$\exists x \varphi \vdash_\sigma \exists x(x = x \wedge \varphi)$$

and

$$\exists x(x = x \wedge \varphi) \vdash_\sigma \exists x \varphi$$

for any σ containing the free variables of $\exists x \varphi$.

Exercise 67. Can you express: $A \xrightarrow{f} B$ is regular mono?

4.4 Example of a regular category

In this section, I treat an example of a type of regular categories which are important in categorical logic. They are categories of Ω-valued sets for some frame Ω. Let's define some things.

Definition 4.12 *A* frame Ω *is a partially ordered set which has suprema (least upper bounds)* $\bigvee B$ *of all subsets* B*, and infima (meets)* $\bigwedge A$ *for* finite *subsets* A *(so, there is a top element* $\top$ *and every pair of elements* x, y *has a meet* $x \wedge y$*), and moreover,* $\wedge$ *distributes over* $\bigvee$*, that is,*

$$x \wedge \bigvee B = \bigvee \{x \wedge b | b \in B\}$$

for $x \in \Omega$*,* $B \subseteq \Omega$*.*

Given a frame Ω we define the category $\mathcal{C}_\Omega$ as follows:
Objects are functions $X \xrightarrow{E_X} \Omega$, X a set;
Maps from (X, E_X) to (Y, E_Y) are functions $X \xrightarrow{f} Y$ satisfying the requirement that $E_X(x) \leq E_Y(f(x))$ for all $x \in X$.
It is easily seen that the identity function satisfies this requirement, and if two composable functions satisfy it, their composition does; so we have a category.

Proposition 4.13 *$\mathcal{C}_\Omega$ is a regular category.*

Proof. Let $\{*\}$ be any one-element set, together with the function which sends $*$ to the top element of Ω. Then $\{*\} \to \Omega$ is a terminal object of $\mathcal{C}_\Omega$.

Given (X, E_X) and (Y, E_Y), a product of the two is the object $(X \times Y, E_{X\times Y})$ where $E_{X\times Y}(x, y)$ is defined as $E_X(x) \wedge E_Y(y)$.

Given two arrows $f, g : (X, E_X) \to (Y, E_Y)$ their equalizer is $(X', E_{X'})$ where $X' \subseteq X$ is $\{x \in X | f(x) = g(x)\}$ and $E_{X'}$ is the restriction of E_X to X'.

This is easily checked, and $\mathcal{C}_\Omega$ is a finitely complete category.

An arrow $f : (X, E_X) \to (Y, E_Y)$ is regular epi if and only if the function f is surjective and for all $y \in Y$, $E_Y(y) = \bigvee f^{-1}(y)$.

For suppose f is such, and $g : (X, E_X) \to (Z, E_Z)$ coequalizes the kernel pair of f. Then $g(x) = g(x')$ whenever $f(x) = f(x')$, and so for all $y \in Y$, since $f(x) = y$ implies $E_X(x) \leq E_Z(g(x))$, we have

$$E_Y(y) = \bigvee\{E_X(x)|f(x) = y\} \leq E_Z(g(x))$$

so there is a unique map $h : (Y, E_Y) \to (Z, E_Z)$ such that $g = hf$; that is f is the coequalizer of its kernel pair.

The proof of the converse is left to you.

Finally we must show that regular epis are stable under pullback. This is an exercise. ∎

Exercise 68. Show that the pullback of
$$\begin{tikzcd} & X \arrow[d, "f"] \\ Y \arrow[r, "g"'] & Z \end{tikzcd}$$
(I suppress reference to the E_X etc.) is (up to isomorphism) the set $\{(x, y)|f(x) = g(y)\}$, with $E(x, y) = E_X(x) \wedge E_Y(y)$; and then, use the distributivity of Ω to show that regular epis are stable under pullback.

Exercise 69. Fill in the other gap in the proof: if $f : (X, E_X) \to (Y, E_Y)$ is a regular epi, then f satisfies the condition given in the proof.

Exercise 70. Given $(X, E_X) \xrightarrow{f} (Y, E_Y)$, give the interpretation of the formula $\exists x(f(x) = y)$, as subobject of (Y, E_Y).

Exercise 71. Characterize those objects (X, E_X) for which the unique map to the terminal object is a regular epimorphism.

Exercise 72. Give a categorical proof of the statement: if f is the coequalizer of something, it is the coequalizer of its kernel pair.

Exercise 73. Characterize the regular monos in $\mathcal{C}_\Omega$.

5 Adjunctions

The following kind of problem occurs quite regularly: suppose we have a functor $\mathcal{C} \xrightarrow{G} \mathcal{D}$ and for a given object D of $\mathcal{D}$, we look for an object $G(C)$ which "best approximates" D, in the sense that there is a map $D \xrightarrow{\eta} G(C)$ with the property that any other map $D \xrightarrow{g} G(C')$ factors uniquely as $G(f)\eta$ for $f : C \to C'$ in $\mathcal{C}$.

We have seen, that if G is the inclusion of Abgp into Grp, the abelianization of a group is an example. Another example is the Čech-Stone compactification in topology: for an arbitrary topological space X one constructs a compact space βX out of it, and a map $X \to \beta X$, such that any continuous map from X to a compact space factors uniquely through this map.

Of course, what we described here is a sort of "right-sided" approximation; the reader can define for himself what the notion for a left-sided approxiamtion would be.

The categorical description of this kind of phenomena goes via the concept of *adjunction*, which this chapter is about.

5.1 Adjoint functors

Let $\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$ be a pair of functors between categories $\mathcal{C}$ and $\mathcal{D}$.

We say that F is *left adjoint* to G, or G is *right adjoint* to F, notation: $F \dashv G$, if there is a natural bijection:

$$\mathcal{C}(FD, C) \xrightarrow{m_{D,C}} \mathcal{D}(D, GC)$$

for each pair of objects $C \in \mathcal{C}_0, D \in \mathcal{D}_0$. Two maps $f : FD \to C$ in $\mathcal{C}$ and $g : D \to GC$ in $\mathcal{D}$ which correspond to each other under this correspondence are called *transposes* of each other.

The naturality means that, given $f : D \to D'$, $g : C' \to C$ in $\mathcal{D}$ and $\mathcal{C}$ respectively, the diagram

$$\begin{tikzcd}
\mathcal{C}(FD, C) \arrow[r, "m_{D,C}"] & \mathcal{D}(D, GC) \\
\mathcal{C}(FD', C') \arrow[u, "\mathcal{C}(Ff,g)"] \arrow[r, "m_{D',C'}"'] & \mathcal{D}(D', GC') \arrow[u, "\mathcal{D}(f,Gg)"']
\end{tikzcd}$$

commutes in Set. Remind yourself that given $\alpha : FD' \to C'$, $\mathcal{C}(Ff, g)(\alpha) : FD \to C$ is the composite

$$FD \xrightarrow{Ff} FD' \xrightarrow{\alpha} C' \xrightarrow{g} C$$

Such a family $m = (m_{D,C} | D \in \mathcal{D}_0, C \in \mathcal{C}_0)$ is then completely determined by the values it takes on identities; i.e. the values

$$m_{D,FD}(\mathrm{id_{FD}}) : \mathrm{D} \to \mathrm{GFD}$$

For, given $\alpha : FD \to C$, since $\alpha = \mathcal{C}(\mathrm{id_{FD}}, \alpha)(\mathrm{id_{FD}})$,

$$\begin{array}{rcl} m_{D,C}(\alpha) & = & m_{D,C}(\mathcal{C}(\mathrm{id_{FD}}, \alpha)(\mathrm{id_{FD}})) \qquad = \\ & & \mathcal{D}(\mathrm{id_D}, \mathrm{G}(\alpha))(\mathrm{m_{D,FD}}(\mathrm{id_{FD}})) \end{array}$$

which is the composite

$$D \xrightarrow{m_{D,FD}(\mathrm{id_{FD}})} GFD \xrightarrow{\quad G(\alpha) \quad} G(C)$$

The standard notation for $m_{D,FD}(\mathrm{id_{FD}})$ is $\eta_D : D \to GF(D)$.

Exercise 74. Show that $(\eta_D : D \in \mathcal{D}_0)$ is a natural transformation:

$$\mathrm{id}_{\mathcal{D}} \Rightarrow \mathrm{GF}$$

By the same reasoning, the natural family $(m^{-1}_{D,C} | D \in \mathcal{D}_0, C \in \mathcal{C}_0)$ is completely determined by its actions on identities

$$m^{-1}_{GC,C}(\mathrm{id_{GC}}) : \mathrm{FGC} \to \mathrm{C}$$

Again, the family $(m^{-1}_{GC,C}(\mathrm{id_{GC}}) | \mathrm{C} \in \mathcal{C}_0)$ is a natural transformation: $FG \Rightarrow \mathrm{id}_{\mathcal{C}}$. We denote its components by ε_C and this is also standard notation.

We have that $m^{-1}_{D,C}(\beta : D \to GC)$ is the composite

$$FD \xrightarrow{F\beta} FGC \xrightarrow{\varepsilon_C} C$$

Now making use of the fact that $m_{D,C}$ and $m^{-1}_{D,C}$ are each others inverse we get that for all $\alpha : FD \to C$ and $\beta : D \to GC$ the diagrams

$$\begin{tikzcd} D \arrow[r, "\beta"] \arrow[d, "\eta_D"'] & GC \\ GFD \arrow[r, "GF(\beta)"'] & GFG(C) \arrow[u, "G(\varepsilon_C)"'] \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} FD \arrow[r, "\alpha"] \arrow[d, "F(\eta_D)"'] & C \\ FGFD \arrow[r, "FG(\alpha)"'] & FGC \arrow[u, "\varepsilon_C"'] \end{tikzcd}$$

commute; applying this to the identities on FD and GC we find that we have commuting diagrams of natural transformations:

$$\begin{tikzcd} G \arrow[r, Rightarrow, "\eta\star G"] \arrow[dr, Rightarrow, "\mathrm{id_G}"'] & GFG \arrow[d, Rightarrow, "G\circ\varepsilon"] \\ & G \end{tikzcd} \qquad \begin{tikzcd} F \arrow[r, Rightarrow, "F\circ\eta"] \arrow[dr, Rightarrow, "\mathrm{id_F}"'] & FGF \arrow[d, Rightarrow, "\varepsilon\star F"] \\ & F \end{tikzcd}$$

Here $\eta \star G$ denotes $(\eta_{GC}|C \in \mathcal{C}_0)$ and $G \circ \varepsilon$ denotes $(G(\varepsilon_C)|C \in \mathcal{C}_0)$.

Conversely, given $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ with natural transformations $\eta : \mathrm{id}_{\mathcal{D}} \Rightarrow \mathrm{GF}$ and $\varepsilon : FG \Rightarrow \mathrm{id}_{\mathcal{C}}$ which satisfy the above *triangle equalities*, we have that $F \dashv G$.

The tuple $(F, G, \varepsilon, \eta)$ is called an *adjunction*. η is the *unit* of the adjunction, ε the *counit*.

Exercise 75. Prove the last statement, i.e. given $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$, $\eta : \mathrm{id}_{\mathcal{D}} \Rightarrow \mathrm{GF}$ and $\varepsilon : FG \Rightarrow \mathrm{id}_{\mathcal{C}}$ satisfying $(G \circ \varepsilon) \cdot (\eta \star G) = \mathrm{id_G}$ and $(\varepsilon \star F) \cdot (F \circ \eta) = \mathrm{id_F}$, we have $F \dashv G$.

Exercise 76. Given $\mathcal{C} \underset{G_1}{\overset{F_1}{\leftrightarrows}} \mathcal{D} \underset{G_2}{\overset{F_2}{\leftrightarrows}} \mathcal{E}$, if $F_1 \dashv G_1$ and $F_2 \dashv G_2$ then $F_1F_2 \dashv G_2G_1$.

Examples. The world is full of examples of adjoint functors. We have met several:

a) Consider the forgetful functor U : Grp $\to$ Set and the free functor F : Set $\to$ Grp. Given a function from a set A to a group G (which is an arrow $A \to U(G)$ in Set) we can uniquely extend it to a group homomorphism from $(\tilde{A}, \star)$ to G (see example e) of 1.1), i.e. an arrow $F(A) \to G$ in Grp, and conversely. This is natural in A and G, so $F \dashv U$;

b) The functor Dgrph $\to$ Cat given in example f) of 1.1 is left adjoint to the forgetful functor Cat $\to$ Dgrph;

c) Given functors $P \underset{G}{\overset{F}{\leftrightarrows}} Q$ between two preorders P and Q, $F \dashv G$ if and only if we have the equivalence

$$y \leq G(x) \Leftrightarrow F(y) \leq x$$

for $x \in P, y \in Q$; if and only if we have $FG(x) \leq x$ and $y \leq GF(y)$ for all x, y;

d) In example m) of 1.1 we did "abelianization" of a group G. We made use of the fact that any homomorphism $G \to H$ with H abelian, factors uniquely through $G/[G,G]$, giving a natural 1-1 correspondence

$$\mathrm{Grp}(G, I(H)) \xrightarrow{\sim} \mathrm{Abgp}(G/[G,G], H)$$

where $I : \mathrm{Abgp} \to \mathrm{Grp}$ is the inclusion. So abelianization is left adjoint to the inclusion functor of abelian groups into groups;

e) The *free monoid* $F(A)$ on a set A is just the set of strings on the alphabet A. $F : \mathrm{Set} \to \mathrm{Mon}$ is a functor left adjoint to the forgetful functor from Mon to Set;

f) Given a set X we have seen (example g) of 2.2) the product functor $(-) \times X : \mathrm{Set} \to \mathrm{Set}$, assigning the product $Y \times X$ to a set Y.

Since there is a natural bijection between functions $Y \times X \to Z$ and functions $Y \to Z^X$, the functor $(-)^X : \mathrm{Set} \to \mathrm{Set}$ is right adjoint to $(-) \times X$;

g) Example e) of 2.2 gives two functors $F, G : \mathrm{Set} \to \mathrm{Cat}$. F and G are respectively left and right adjoint to the functor $\mathrm{Cat} \xrightarrow{\mathrm{Ob}} \mathrm{Set}$ which assigns to a (small) category its set of objects (to be precise, for this example to work we have to take for Cat the category of *small* categories), and to a functor its action on objects.

h) Given a regular category $\mathcal{C}$ we saw in 4.1 that every arrow $f : X \to Y$ can be factored as a regular epi followed by a monomorphism. This gives rise to a monotone function: $\mathrm{Sub}(X) \xrightarrow{\mathrm{Im}_f} \mathrm{Sub}(Y)$ defined as follows: if $M \in \mathrm{Sub}(X)$ is represented by m, $\mathrm{Im}_f(M)$ is the image of fm (see 4.1). We have seen that $\mathrm{Im}_f(M) \leq N \Leftrightarrow M \leq f^*(N)$ for any subobject N of Y, so Im_f is left adjoint to f^*.

We can also express this logically: in the logic developed in chapter 4, for any formulas $M(x)$ and $N(y)$, the sequents

$$\exists x(f(x) = y \wedge M(x)) \vdash_y N(y)$$

and

$$M(x) \vdash_x N(f(x))$$

are equivalent.

One of the slogans of categorical logic is therefore, that "existential quantification is left adjoint to substitution".

i) Let $\mathcal{C}$ be a category with finite products; for $C \in \mathcal{C}_0$ consider the slice category $\mathcal{C}/C$. There is a functor $C^* : \mathcal{C} \to \mathcal{C}/C$ which assigns to D the object $C \times D \stackrel{\pi_C}{\to} C$ of $\mathcal{C}/C$, and to maps $D \stackrel{f}{\to} D'$ the map $\mathrm{id_C} \times \mathrm{f}$.

   C^* has a left adjoint Σ_C which takes the domain: $\Sigma_C(D \to C) = D$.

j) Let $P : \mathrm{Set}^{\mathrm{op}} \to \mathrm{Set}$ be the functor which takes the powerset on objects, and for $X \stackrel{f}{\to} Y$, $P(f) : P(Y) \to P(X)$ gives for each subset B of Y its inverse image under f.

   Now P might as well be regarded as a functor $\mathrm{Set} \to \mathrm{Set}^{\mathrm{op}}$; let's write $\bar{P}$ for that functor. Since there is a natural bijection:

$$\mathrm{Set}(X, P(Y)) \stackrel{\sim}{\to} \mathrm{Set}(Y, P(X)) = \mathrm{Set}^{\mathrm{op}}(\bar{P}(X), Y)$$

   we have an adjunction $\bar{P} \dashv P$.

Exercise 77. Suppose that $\mathcal{C} \stackrel{F}{\leftarrow} \mathcal{D}$ is a functor and that for each object C of $\mathcal{C}$ there is an object GC of $\mathcal{D}$ and an arrow $\varepsilon_C : FGC \to C$ with the property that for every object D of $\mathcal{D}$ and any map $FD \stackrel{f}{\to} C$, there is a unique $\tilde{f} : D \to GC$ such that

$$\begin{tikzcd} FD \arrow[rr, "f"] \arrow[dr, "F\tilde{f}"'] & & C \\ & FGC \arrow[ur, "\varepsilon_C"'] & \end{tikzcd}$$

commutes.

Prove that $G : \mathcal{C}_0 \to \mathcal{D}_0$ extends to a functor $G : \mathcal{C} \to \mathcal{D}$ which is right adjoint to F, and that $(\varepsilon_C : FGC \to C | C \in \mathcal{C}_0)$ is the counit of the adjunction.

Construct also the unit of the adjunction.

Exercise 78. Given $\mathcal{C} \stackrel{G}{\to} \mathcal{D}$, for each object D of $\mathcal{D}$ we let $(D{\downarrow}G)$ denote the category which has as objects pairs (C, g) where C is an object in $\mathcal{C}$ and $g : D \to GC$ is an arrow in $\mathcal{D}$. An arrow $(C, g) \to (C', g')$ in $(D{\downarrow}G)$ is an arrow $f : C \to C'$ in $\mathcal{C}$ which makes

$$\begin{tikzcd} & D \arrow[dl, "g"'] \arrow[dr, "g'"] & \\ GC \arrow[rr, "Gf"'] & & GC' \end{tikzcd}$$

commute.

Show that G has a left adjoint if and only if for each D, the category $(D{\downarrow}G)$ has an initial object.

5.2 Expressing (co)completeness by existence of adjoints; preservation of (co)limits by adjoint functors

Given categories $\mathcal{C}$ and $\mathcal{D}$, we defined for every functor $F : \mathcal{C} \to \mathcal{D}$ its *limit* (or limiting cone), if it existed, as a pair (D, μ) with $\mu : \Delta_D \Rightarrow F$, and (D, μ) terminal in the category of cones for F.

Any other natural transformation $\mu' : \Delta_{D'} \Rightarrow F$ factors uniquely through (D, μ) via an arrow $D' \to D$ in $\mathcal{D}$ and conversely, every arrow $D' \to D$ gives rise to a natural transformation $\mu' : \Delta_{D'} \Rightarrow F$.

So there is a 1-1 correspondence between

$$\mathcal{D}(D', D) \text{ and } \mathcal{D}^{\mathcal{C}}(\Delta_{D'}, F)$$

which is natural in D'.

Since every arrow $D' \to D''$ in $\mathcal{D}$ gives a natural transformation $\Delta_{D'} \Rightarrow \Delta_{D''}$ (example i) of 2.2), there is a functor $\Delta_{(-)} : \mathcal{D} \to \mathcal{D}^{\mathcal{C}}$.

The above equation now means that:

Proposition 5.1 *$\mathcal{D}$ has all limits of type $\mathcal{C}$ (i.e. every functor $\mathcal{C} \xrightarrow{F} \mathcal{D}$ has a limiting cone in $\mathcal{D}$) if and only if $\Delta_{(-)}$ has a right adjoint.*

Exercise 79. Give an exact proof of this proposition.

Exercise 80. Use duality to deduce the dual of the proposition: $\mathcal{D}$ has all colimits of type $\mathcal{C}$ if and only if $\Delta_{(-)} : \mathcal{D} \to \mathcal{D}^{\mathcal{C}}$ has a left adjoint.

Exercise 81. (Uniqueness of adjoints) Any two left (or right) adjoints of a given functor are isomorphic as objects of the appropriate functor category.

Exercise 82. $\mathcal{D} \to \mathbf{1}$ has a right adjoint iff $\mathcal{D}$ has a terminal object, and a left adjoint iff $\mathcal{D}$ has an initial object.

Exercise 83. Suppose $\mathcal{D}$ has both an initial and a terminal object; denote by L the functor $\mathcal{D} \to \mathcal{D}$ which sends everything to the initial, and by R the one which sends everything to the terminal object. $L \dashv R$.

Exercise 84. Let $F \dashv G$ with counit $\varepsilon : FG \Rightarrow \mathrm{id}$. Show that ε is a natural isomorphism if and only if G is faithful.

A very important aspect of adjoint functors is their behaviour with respect to limits and colimits.

Theorem 5.2 *Let* $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ *such that $F \dashv G$. Then:*

a) *G preserves all limits which exist in $\mathcal{C}$;*

b) *F preserves all colimits which exist in $\mathcal{D}$.*

Proof. Suppose $M : \mathcal{E} \to \mathcal{C}$ has a limiting cone (C, μ) in $\mathcal{C}$. Now a cone (D, ν) for GM is a natural family $D \xrightarrow{\nu_E} GM(E)$, i.e. such that

$$\begin{tikzcd} D \arrow[r, "\nu_E"] \arrow[dr, "\nu_{E'}"'] & GM(E) \arrow[d, "GM(e)"] \\ & GM(E') \end{tikzcd}$$

commutes for every $E \xrightarrow{e} E'$ in $\mathcal{E}$.

This transposes under the adjunction to a family $(FD \xrightarrow{\tilde{\nu}_E} ME | E \in \mathcal{E}_0)$ and the naturality requirement implies that

$$\begin{tikzcd} FD \arrow[r, "\tilde{\nu}_E"] \arrow[dr, "\tilde{\nu}_{E'}"'] & ME \arrow[d, "M(e)"] \\ & ME' \end{tikzcd}$$

commutes in $\mathcal{C}$, in other words, that (FD, ν) is a cone for M in $\mathcal{C}$. There is, therefore, a unique map of cones from $(FD, \tilde{\nu})$ to (C, μ).

Transposing back again, we get a unique map of cones $(D, \nu) \to (GC, G \circ \mu)$. That is to say that $(GC, G \circ \mu)$ is terminal in Cone(GM), so a limiting cone for GM, which was to be proved.

The argument for the other statement is dual. ∎

Exercise 85. Given $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$, $F \dashv G$ and $M : \mathcal{E} \to \mathcal{C}$. Show that the functor Cone$(M) \to$ Cone(GM) induced by G has a left adjoint.

From the theorem on preservation of (co)limits by adjoint functors one can often conclude that certain functors cannot have a right or a left adjoint.

Examples

a) The forgetful functor Mon $\to$ Set does not preserve epis, as we have seen in 1.2. In chapter 3 we've seen that f is epi iff $\begin{tikzcd} \cdot \arrow[r, "\mathrm{id}"] \arrow[d, "\mathrm{id}"'] & \cdot \arrow[d, "f"] \\ \cdot \arrow[r, "f"'] & \cdot \end{tikzcd}$ is a pushout; since left adjoints preserve identities and pushouts, they preserve epis; therefore the forgetful functor Mon $\to$ Set does not have a right adjoint;

b) The functor $(-)\times X : \mathrm{Set} \to \mathrm{Set}$ does not preserve the terminal object unless X is itself terminal in Set; therefore, it does not have a left adjoint for non-terminal X.

c) The forgetful functor Pos $\to$ Set has a left adjoint, but it cannot have a right adjoint: it preserves all coproducts, including the initial object, but not all coequalizers.

Exercise 86. Prove the last example. Hint: think of the coequalizer of the following two maps $\mathbb{Q} \to \mathbb{R}$: one is the inclusion, the other is the constant zero map.

Another use of the theorem has to do with the computation of limits. Many categories, as we have seen, have a forgetful functor to Set which has a left adjoint. So the forgetful functor preserves limits, and since these can easily be computed in Set, one already knows the "underlying set" of the vertex of the limiting cone one wants to compute.

Does a converse to the theorem hold? I.e. given $G : \mathcal{C} \to \mathcal{D}$ which preserves all limits; does G have a left adjoint? In general *no*, unless $\mathcal{C}$ is sufficiently complete, and a rather technical condition, the "solution set condition" holds. The *adjoint functor theorem* (Freyd) tells that in that case there is a converse:

Definition 5.3 (Solution set condition) *$G : \mathcal{C} \to \mathcal{D}$ satisfies the solution set condition (ssc) for an object D of $\mathcal{D}$, if there is a* set *X_D of objects of $\mathcal{C}$, such that every arrow $D \to GC$ factors as*

$$\begin{tikzcd} D \arrow[r] \arrow[dr] & GC \\ & GC' \arrow[u, "G(f)"'] \end{tikzcd}$$

for some $C' \in X_D$ and $f : C' \to C$ in $\mathcal{C}$.

Theorem 5.4 (Adjoint Functor Theorem) *Let $\mathcal{C}$ be a complete category and $G : \mathcal{C} \to \mathcal{D}$ a functor. G has a left adjoint if and only if G preserves all small limits and satisfies the ssc for every object D of $\mathcal{D}$.*

I won't prove the theorem, but you may like to try yourself. It is a, not too trivial, exercise.

For small categories $\mathcal{C}$, the ssc is of course irrelevant. But categories which are small *and* complete are rather special...they are complete preorders.

For preorders $\mathcal{C}$, $\mathcal{D}$ we have: if $\mathcal{C}$ is complete, then $G : \mathcal{C} \to \mathcal{D}$ has a left adjoint if and only if G preserves all limits, that is: greatest lower bounds $\bigwedge B$ for all $B \subseteq \mathcal{C}$. For, put

$$F(d) = \bigwedge\{c | d \leq G(c)\}$$

Then $F(d) \leq c'$ implies (since G preserves $\bigwedge$) $\bigwedge\{G(c)|d \leq G(c)\} \leq G(c')$ which implies $d \leq G(c')$ since $d \leq \bigwedge\{G(c)|d \leq G(c)\}$; conversely, $d \leq G(c')$ implies $c' \in \{c|d \leq G(c)\}$ so $F(d) = \bigwedge\{c|d \leq G(c)\} \leq c'$.

6 Monads and Algebras

Given an adjunction $(F, G, \varepsilon, \eta) : \mathcal{C} \leftrightarrows \mathcal{D}$ let us look at the functor $T = GF : \mathcal{D} \to \mathcal{D}$.

We have a natural transformation $\eta : \mathrm{id}_{\mathcal{D}} \Rightarrow \mathrm{T}$ and a natural transformation $\mu : T^2 \Rightarrow T$. The components μ_D are

$$T^2(D) = GFGFD \overset{G(\varepsilon_{FD})}{\to} GFD = T(D)$$

Furthermore the equalities

$$\begin{tikzcd} T^3 \arrow[r, "T\mu"] \arrow[d, "\mu T"'] & T^2 \arrow[d, "\mu"] \\ T^2 \arrow[r, "\mu"'] & T \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} T \arrow[r, "\eta T"] \arrow[dr, "="'] & T^2 \arrow[d, "\mu"] & T \arrow[l, "T\eta"'] \arrow[dl, "="] \\ & T & \end{tikzcd}$$

hold. Here $(T\mu)_D = T(\mu_D) : T^3D \to TD$ and $(\mu T)_D = \mu_{TD} : T^3D \to TD$ (Similar for ηT and $T\eta$).

Exercise 87. Prove these equalities.

A triple (T, μ, η) satisfying these identities is called a *monad*. Try to see the formal analogy between the defining equalities for a monad and the axioms for a monoid: writing $m(e, f)$ for ef in a monoid, and η for the unit element, we have

$$\begin{array}{ll} m(e, m(g, h)) = m(m(e, g), h) & \text{(associativity)} \\ m(\eta, e) = m(e, \eta) = e & \text{(unit)} \end{array}$$

Following this one calls μ the *multiplication* of the monad, and η its *unit*.

Example. The powerset functor $\mathcal{P}$: Set $\to$ Set (example j) of 2.2, with η and μ indicated there) is a monad (check).

Dually, there is the notion of a *comonad* (L, δ, ε) on a category $\mathcal{C}$, with equalities

$$\begin{tikzcd} L \arrow[r, "\delta"] \arrow[d, "\delta"'] & L^2 \arrow[d, "L\delta"] \\ L^2 \arrow[r, "\delta L"'] & L^3 \end{tikzcd} \qquad \begin{tikzcd} & L \arrow[dl, "="'] \arrow[d, "\delta"] \arrow[dr, "="] & \\ L & L^2 \arrow[l, "\varepsilon L"] \arrow[r, "L\varepsilon"'] & L \end{tikzcd}$$

Given an adjunction $(F, G, \varepsilon, \eta)$, $(FG, \delta = F\eta G, \varepsilon)$ is a comonad on $\mathcal{C}$. We call δ the *comultiplication* and ε the *counit* (this is in harmony with the unit-counit terminology for adjunctions).

Although, in many contexts, comonads and the notions derived from them are at least as important as monads, the treatment is dual so I concentrate on monads.

Every adjunction gives rise to a monad; conversely, every monad arises from an adjunction, but in more than one way. Essentially, there are a maximal and a minimal solution to the problem of finding an adjunction from which a given monad arises.

Example. A monad on a poset P is a monotone function $T : P \to P$ with the properties $x \leq T(x)$ and $T^2(x) \leq T(x)$ for all $x \in P$; such an operation is also often called a *closure operation* on P. Note that $T^2 = T$ because T is monotone.

In this situation, let $Q \subseteq P$ be the image of T, with the ordering inherited from P. We have maps $r : P \to Q$ and $i : Q \to P$ such that ri is the identity on Q and $ir = T : P \to P$.

For $x \in P$, $y \in Q$ we have $x \leq i(y) \Leftrightarrow r(x) \leq y$ (check); so $r \dashv i$ and the operation T arises from this adjunction.

6.1 Algebras for a monad

Given a monad (T, η, μ) on a category $\mathcal{C}$, we define the category T-Alg of T-*algebras* as follows:

- *Objects* are pairs (X, h) where X is an object of $\mathcal{C}$ and $h : T(X) \to X$ is an arrow in $\mathcal{C}$ such that

$$\begin{tikzcd} T^2(X) \arrow[r, "T(h)"] \arrow[d, "\mu_X"'] & T(X) \arrow[d, "h"] \\ T(X) \arrow[r, "h"'] & X \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[dr, "="'] & T(X) \arrow[d, "h"] \\ & X \end{tikzcd}$$

  commute;

- *Morphisms*: $(X, h) \to (Y, k)$ are morphisms $X \xrightarrow{f} Y$ in $\mathcal{C}$ for which

$$\begin{tikzcd} T(X) \arrow[r, "T(f)"] \arrow[d, "h"'] & T(Y) \arrow[d, "k"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

  commutes.

Theorem 6.1 *There is an adjunction between T-Alg and $\mathcal{C}$ which brings about the given monad T.*

Proof. There is an obvious forgetful functor $U^T : T\text{-Alg} \to \mathcal{C}$ which takes (X, h) to X. I claim that U^T has a left adjoint F^T:

F^T assigns to an object X the T-algebra $T^2(X) \xrightarrow{\mu_X} T(X)$; to $X \xrightarrow{f} Y$ the map $T(f)$; this is an algebra map because of the naturality of μ. That $T^2(X) \xrightarrow{\mu_X} T(X)$ is an algebra follows from the defining axioms for a monad T.

Now given any arrow $g : X \to U^T(Y, h)$ we let $\tilde{g} : (T(X), \mu_X) \to (Y, h)$ be the arrow $T(X) \xrightarrow{T(g)} T(Y) \xrightarrow{h} Y$. This is a map of algebras since

$$\begin{tikzcd}
T^2(X) \arrow[r, "T^2(g)"] \arrow[d, "\mu_X"'] & T^2(Y) \arrow[r, "T(h)"] \arrow[d, "\mu_Y"'] & Y \arrow[d, "h"] \\
T(X) \arrow[r, "T(g)"'] & T(Y) \arrow[r, "h"'] & Y
\end{tikzcd}$$

commutes; the left hand square is the naturality of μ; the right hand square is because (Y, h) is a T-algebra.

Conversely, given $f : (TX, \mu_X) \to (Y, h)$ we have an arrow $\bar{f} : X \to Y$ by taking the composite $X \xrightarrow{\eta_X} TX \xrightarrow{f} Y$.

Now $\tilde{\bar{f}} : TX \to Y$ is the composite

$$TX \xrightarrow{T(\eta_X)} T^2X \xrightarrow{T(f)} TY \xrightarrow{h} Y$$

By naturality of η this is

$$TX \xrightarrow{f} Y \xrightarrow{\eta_Y} TY \xrightarrow{h} Y$$

which is f since (Y, h) is a T-algebra.

Conversely, $\bar{\tilde{g}} : X \to Y$ is the composite

$$X \xrightarrow{\eta_X} TX \xrightarrow{T(g)} TY \xrightarrow{h} Y$$

Again by naturality of η and the fact that (Y, h) is a T-algebra, we conclude that $\bar{\tilde{g}} = g$. So we have a natural 1-1 correspondence

$$\mathcal{C}(X, U^T(Y, h)) \simeq T\text{-Alg}(F^T(X), (Y, h))$$

and our adjunction.

Note that the composite $U^T F^T$ is the functor T, and that the unit η of the adjunction is the unit of T; the proof that for the counit ε of $F^T \dashv U^T$ we have that

$$T^2 = U^T F^T U^T F^T \xrightarrow{U^T \varepsilon F^T} U^T F^T = T$$

is the original multiplication μ, is left to you. ∎

Exercise 88. Complete the proof.

Example. The *group monad*. Combining the forgetful functor $U : \mathrm{Grp} \to \mathrm{Set}$ with the left adjoint, the free functor $\mathrm{Set} \to \mathrm{Grp}$, we get the following monad on Set:

$T(A)$ is the set of sequences on the alphabet $A \sqcup A^{-1}$ (A^{-1} is the set $\{a^{-1} | a \in A\}$ of formal inverses of elements of A, as in example e) of 1.1) in which no aa^{-1} or $a^{-1}a$ occur. The unit $A \stackrel{\eta_A}{\to} TA$ sends $a \in A$ to the string $\langle a \rangle$. The multiplication $\mu : T^2(A) \to T(A)$ works as follows. Define $(-)^- : A \sqcup A^{-1} \to A \sqcup A^{-1}$ by $a^- = a^{-1}$ and $(a^{-1})^- = a$. Define also $(-)^-$ on strings by $(a_1 \ldots a_n)^- = a_n^- \ldots a_1^-$. Now for an element of $TT(A)$, which is a string on the alphabet $T(A) \sqcup T(A)^{-1}$, say $\sigma_1 \ldots \sigma_n$, we let $\mu_A(\sigma_1 \ldots \sigma_n)$ be the concatenation of the strings $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n$ on the alphabet $A \sqcup A^{-1}$, where $\tilde{\sigma}_i = \sigma_i$ if $\sigma_i \in T(A)$, and $\tilde{\sigma}_i = (\sigma_i)^-$ if $\sigma_i \in T(A)^{-1}$. Of course we still have to remove possible substrings of the form aa^{-1} etc.

Now let us look at algebras for the group monad: maps $T(A) \stackrel{h}{\to} A$ such that for a string of strings

$$\alpha = \sigma_1, \ldots, \sigma_n = \langle \langle s_1^1, \ldots, s_1^{k_1} \rangle, \ldots, \langle s_n^1, \ldots, s_n^{k_n} \rangle \rangle$$

we have that

$$h(\langle h\sigma_1, \ldots, h\sigma_n \rangle) = h(\langle s_1^1, \ldots, s_1^{k_1}, \ldots, s_n^1, \ldots, s_n^{k_n} \rangle)$$

and

$$h(\langle a \rangle) = a \text{ for } a \in A$$

I claim that this is the same thing as a group structure on A, with multiplication $a \cdot b = h(\langle a, b \rangle)$.

The unit element is given by $h(\langle\rangle)$; the inverse of $a \in A$ is $h(\langle a^{-1} \rangle)$ since

$$\begin{array}{rcl} h(\langle a, h(\langle a^{-1} \rangle) \rangle) & = & h(\langle h(\langle a \rangle), h(\langle a^{-1} \rangle) \rangle) = \\ h(\langle a, a^{-1} \rangle) & = & h(\langle\rangle), \text{ the unit element} \end{array}$$

Try to see for yourself how the associativity of the monad and its algebras transforms into associativity of the group law.

Exercise 89. Finish the proof of the theorem: for the group monad T, there is an equivalence of categories between T-Alg and Grp.

This situation is very important and has its own name:

Definition 6.2 *Given an adjunction* $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ *,* $F \dashv G$ *there is always a* comparison functor $K : \mathcal{C} \to T\text{-}Alg$ *for* $T = GF$*, the monad induced by the adjunction.* K *sends an object* C *of* $\mathcal{C}$ *to the* T*-algebra* $GFG(C) \overset{G(\varepsilon_C)}{\to} G(C)$*.*

We say that $\mathcal{C}$ *is* monadic *over* $\mathcal{D}$ *if* K *is an equivalence.*

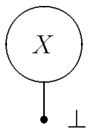
Exercise 90. Check that $K(C)$ is a T-algebra. Complete the definition of K as a functor. Check that in the example of the group monad, the functor T-Alg $\to$ Grp defined there is a pseudo inverse to the comparison functor K.

In many cases however, the situation is not monadic. Take the forgetful functor U : Pos $\to$ Set. It has a left adjoint F which sends a set X to the discrete ordering on X ($x \le y$ iff $x = y$). Of course, UF is the identity on Set and the UF-algebras are just sets. The comparison functor K is the functor U, and this is not an equivalence.

Exercise 91. Why not? [Hint: think of coproducts in both categories. Every equivalence reflects coproducts]

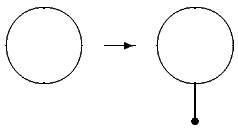
Another example of a monadic situation is of importance in domain theory. Let Pos$_\perp$ be the category of partially ordered sets with a least element, and order preserving maps which also preserve the least element.

There is an obvious inclusion functor U : Pos$_\perp$ $\to$ Pos, and U has a left adjoint F. Given a poset X, $F(X)$ is X "with a bottom element added":

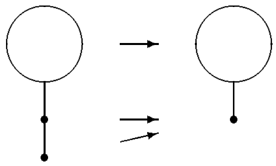


Given $f : X \to Y$ in Pos, $F(f)$ sends the new bottom element of X to the new bottom element of Y, and is just f on the rest. If $f : X \to Y$ in Pos is a map and Y has a least element, we have $F(X) \to Y$ in Pos$_\perp$ by sending $\perp$ to the least element of Y. So the adjunction is clear.

The monad UF : Pos $\to$ Pos, just adding a least element, is called the *lifting monad.* Unit and multiplication are:



$\eta_X : X \to T(X)$



$\mu_X : T^2(X) \to T(X)$

A T-algebra $h : TX \to X$ is first of all a monotone map, but since $h\eta_X = \mathrm{id}_X$, h is epi in Pos so surjective. It follows that X must have a least element $h(\perp)$. From the axioms for an algebra one deduces that h must be the identity when restricted to X, and $h(\perp)$ the least element of X.

Exercise 92. Given $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$, $F \dashv G$, $T = GF$. Prove that the comparison functor $K : \mathcal{C} \to T\text{-Alg}$ satisfies $U^T K = G$ and $KF = F^T$ where $T\text{-Alg} \underset{U^T}{\overset{F^T}{\leftrightarrows}} \mathcal{D}$ as in theorem 6.1.

Another poset example: algebras for the power set monad $\mathcal{P}$ on Set (example j);2.2). Such an algebra $h : \mathcal{P}(X) \to X$ must satisfy $h(\{x\}) = x$ and for $\alpha \subseteq \mathcal{P}(X)$:

$$h(\{h(A) | A \in \alpha\}) = h(\{x | \exists A \in \alpha (x \in A)\}) = h(\bigcup \alpha)$$

Now we can, given an algebra structure on X, define a partial order on X by putting $x \leq y$ iff $h(\{x, y\}) = y$.

Indeed, this is clearly reflexive and antisymmetric. As to transitivity, if $x \leq y$ and $y \leq z$ then

$$\begin{aligned} h(\{x,z\}) &= h(\{x, h(\{y,z\})\}) &= \\ h(\{h(\{x\}), h(\{y,z\})\}) &= h(\{x\} \cup \{y,z\}) &= \\ h(\{x,y\} \cup \{z\}) &= h(\{h(\{x,y\}), h(\{z\})\}) &= \\ h(\{y,z\}) &= z \end{aligned}$$

so $x \leq z$.

Furthermore this partial order is *complete*: least upper bounds for arbitrary subsets exist. For $\bigvee B = h(B)$ for $B \subseteq X$: for $x \in B$ we have $h(\{x, h(B)\}) = h(\{x\} \cup B) = h(B)$ so $x \leq \bigvee B$; and if $x \leq y$ for all $x \in B$ then

$$\begin{aligned} h(\{h(B), y\}) &= h(B \cup \{y\}) &= \\ h(\textstyle\bigcup_{x \in B} \{x, y\}) &= h(\{h(\{x,y\}) | x \in B\}) &= \\ h(\{y\}) &= y \end{aligned}$$

so $\bigvee B \leq y$.

We can also check that a map of algebras is a $\bigvee$-preserving monotone function. Conversely, every $\bigvee$-preserving monotone function between complete posets determines a $\mathcal{P}$-algebra homomorphism.

We have an equivalence between the category of complete posets and $\bigvee$-preserving functions, and $\mathcal{P}$-algebras.

Exercise 93. Let $P : \mathrm{Set}^{\mathrm{op}} \to \mathrm{Set}$ be the contravariant powerset functor, and $\bar{P}$ its left adjoint, as in j) of 5.1. Let $T : \mathrm{Set} \to \mathrm{Set}$ the induced monad.

a) Describe unit and multiplication of this monad explicitly.

b) Show that $\mathrm{Set}^{\mathrm{op}}$ is equivalent to T-Alg [Hint: if this proves hard, have a look at VI.4.3 of Johnstone's "Stone Spaces"].

c) Conclude that there is an adjunction

$$\mathrm{CABool} \rightleftarrows \mathrm{Set}$$

which presents CABool as monadic over Set.

6.2 T-Algebras at least as complete as $\mathcal{D}$

Let T be a monad on $\mathcal{D}$. The following exercise is meant to show that if $\mathcal{D}$ has all limits of a certain type, so does T-Alg. In particular, if $\mathcal{D}$ is complete, so is T-Alg; this is often an important application of a monadic situation.

Exercise 94. Let $\mathcal{E}$ be a category such that every functor $M : \mathcal{E} \to \mathcal{D}$ has a limiting cone. Now suppose $M : \mathcal{E} \to T\text{-Alg}$. For objects E of $\mathcal{E}$, let $M(E)$ be the T-algebra $T(m_E) \xrightarrow{h_E} m_E$.

a) Let $(D, (\nu_E | E \in \mathcal{E}_0))$ be a limiting cone for $U^T M : \mathcal{E} \to \mathcal{D}$. Using the T-algebra structure on $M(E)$ and the fact that $U^T M(E) = m_E$, show that there is also a cone $(TD, (\pi_E | E \in \mathcal{E}_0))$ for $U^T M$;

b) Show that the unique map of cones: $(TD, \pi) \to (D, \nu)$ induces a T-algebra structure $TD \xrightarrow{h} D$ on D;

c) Show that $TD \xrightarrow{h} D$ is the vertex of a limiting cone for M in T-Alg.

6.3 The Kleisli category of a monad

I said before that for a monad T on a category $\mathcal{D}$, there are a "maximal and a minimal solution" to the problem of finding an adjunction which induces the given monad.

We've seen the category T-Alg, which we now write as $\mathcal{D}^T$; we also write $G^T : T\text{-Alg} \to \mathcal{D}$ for the forgetful functor. In case T arises from an adjunction $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$, there was a comparison functor $\mathcal{C} \xrightarrow{K} \mathcal{D}^T$. In the diagram

$$\begin{tikzcd} \mathcal{C} \arrow[rr, "K"] \arrow[dr, shift left, "G"] & & \mathcal{D}^T \arrow[dl, shift left, "G^T"] \\ & \mathcal{D} \arrow[ul, shift left, "F"] \arrow[ur, shift left, "F^T"] & \end{tikzcd}$$

we have that $KF = F^T$ and $G^T K = G$.

Moreover, the functor K is unique with this property.

This leads us to define a category T-Adj of adjunctions $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ such that $GF = T$. A *map* of such T-adjunctions from $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ to $\mathcal{C}' \underset{G'}{\overset{F'}{\leftrightarrows}} \mathcal{D}$ is a functor $K : \mathcal{C} \to \mathcal{C}'$ satisfying $KF = F'$ and $G'K = G$.

What we have proved about T-Alg can be summarized by saying that the adjunction $\mathcal{D}^T \underset{G^T}{\overset{F^T}{\leftrightarrows}} \mathcal{D}$ is a *terminal object* in T-Adj. This was the "maximal" solution.

T-Adj has also an initial object: the *Kleisli category* of T, called $\mathcal{D}_T$. $\mathcal{D}_T$ has the same objects as $\mathcal{D}$, but a map in $\mathcal{D}_T$ from X to Y is an arrow $X \xrightarrow{f} T(Y)$ in $\mathcal{D}$. Composition is defined as follows: given $X \xrightarrow{f} T(Y)$ and $Y \xrightarrow{g} T(Z)$ in $\mathcal{D}$, considered as a composable pair of morphisms in $\mathcal{D}_T$, the composition gf in $\mathcal{D}_T$ is the composite

$$X \xrightarrow{f} T(Y) \xrightarrow{T(g)} T^2(Z) \xrightarrow{\mu_Z} T(Z)$$

in $\mathcal{D}$.

Exercise 95. Prove that composition is associative. What are the identities of $\mathcal{D}_T$?

The adjunction $\mathcal{D}_T \underset{G_T}{\overset{F_T}{\leftrightarrows}} \mathcal{D}$ is defined as follows: the functor G_T sends the object X to $T(X)$ and $f : X \to Y$ ($f : X \to T(Y)$ in $\mathcal{D}$) to

$$T(X) \xrightarrow{T(f)} T^2(Y) \xrightarrow{\mu_Y} T(Y)$$

The functor F_T is the identity on objects and sends $X \xrightarrow{f} Y$ to $X \xrightarrow{f} Y \xrightarrow{\eta_Y} T(Y)$, considered as $X \to Y$ in $\mathcal{D}_T$.

Exercise 96. Define unit and counit; check $F_T \dashv G_T$.

Now for every adjunction $\mathcal{C} \underset{G}{\overset{F}{\leftrightarrows}} \mathcal{D}$ with $GF = T$, there is a unique comparison functor $L : \mathcal{D}_T \to \mathcal{C}$ such that $GL = G_T$ and $LF_T = F$.

L sends the object X to $F(X)$ and $f : X \to Y$ (so $f : X \to T(Y) = GF(Y)$ in $\mathcal{D}$) to its transpose $\tilde{f} : F(X) \to F(Y)$.

Exercise 97. Check the commutations. Prove the uniqueness of L w.r.t. these properties.

7 Cartesian closed categories and the λ-calculus

Many set-theoretical constructions are completely determined (up to isomorphism, as always) by their categorical properties in Set. We are therefore tempted to generalize them to arbitrary categories, by taking the characteristic categorical property as a definition. Of course, this procedure is not really well-defined and it requires sometimes a real insight to pick the 'right' categorical generalization. For example, the category of sets has very special properties:

- $f : X \to Y$ is mono if and only if $fg = fh$ implies $g = h$ for any two maps $g, h : 1 \to X$, where 1 is a terminal object (we say 1 is *a generator*);
- objects X and Y are isomorphic if there exist monos $f : X \to Y$ and $g : Y \to X$ (the Cantor-Bernstein theorem);
- every mono $X \xrightarrow{f} Y$ is part of a coproduct diagram

$$\begin{tikzcd} X \arrow[dr, "f"] & \\ Z \arrow[r, "g"'] & Y \end{tikzcd}$$

And if you believe the axiom of choice, there is its categorical version:

- Every epi is split

None of these properties is generally valid, and categorical generalizations based on them are usually of limited value[2].

In this chapter we focus on a categorical generalization of a set-theoretical concept which has proved to have numerous applications: Cartesian closed categories as the generalization of "function space".

In example f) of 5.1 we saw that the set of functions Z^X appears as the value at Z of the right adjoint to the product functor $(-) \times X$. A category is called *cartesian closed* if such right adjoints always exist. In such categories we may really think of this right adjoint as giving the "object of functions (or arrows)", as the treatment of the λ-calculus will make clear.

7.1 Cartesian closed categories (ccc's); examples and basic facts

Definition 7.1 *A category $\mathcal{C}$ is called* cartesian closed *or* a ccc *if it has finite products, and for every object X of $\mathcal{C}$ the product functor $(-) \times X$ has a right adjoint.*

[2] Asperti, for example, restricts his interpretation of the λ-calculus to ccc's where 1 is a generator. You might as well immediately restrict to Set, then.

Of course, "the" product functor only exists once we have chosen a product diagram for every pair of objects of $\mathcal{C}$. In this chapter we assume that we have such a choice, as well as a distinguished terminal object 1; and we assume also that for each object X we have a *specified* right adjoint to the functor $(-) \times X$, which we write as $(-)^X$ (Many authors write $X \Rightarrow (-)$, but I think that overloads the arrows notation too much). Objects of the form Z^X are called *exponents*.

We have the *unit*

$$Y \stackrel{\eta_{Y,X}}{\rightarrow} (Y \times X)^X$$

and *counit*

$$Y^X \times X \stackrel{\varepsilon_{Y,X}}{\rightarrow} Y$$

of the adjunction $(-) \times X \dashv (-)^X$. Anticipating the view of Y^X as the object of arrows $X \rightarrow Y$, we call ε *evaluation*.

Examples

a) A preorder (or partial order) is cartesian closed if it has a top element 1, binary meets $x \wedge y$ and for any two elements x, y an element $x \rightarrow y$ satisfying for each z:

$$z \leq x \rightarrow y \text{ iff } z \wedge x \leq y$$

b) Set is cartesian closed; Cat is cartesian closed (2.1);

c) Top is not cartesian closed, however the category of locally compact spaces and continuous maps is;

d) Pos is cartesian closed. The exponent Y^X is the set of all monotone maps $X \rightarrow Y$, ordered pointwise ($f \leq g$ iff for all $x \in X$, $fx \leq gx$ in Y);

e) Grp and Abgp are not cartesian closed. In both categories, the initial object is the one-element group. Since for non-initial groups G, $(-) \times G$ does not preserve the initial object, it cannot have a right adjoint;

f) **1** is cartesian closed; **0** isn't (why?);

g) $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$ is cartesian closed. Products and 1 are given "pointwise" (in fact all limits are), that is $F \times G(C) = F(C) \times G(C)$ and $1(C)$ is the terminal 1 in Set, for all $C \in \mathcal{C}_0$.

   The construction of the exponent G^F is a nice application of the Yoneda lemma. Indeed, for G^F to be the right adjoint (at G) of $(-) \times F$, we need for every object C of $\mathcal{C}$:

$$\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C \times F, G) \simeq \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C, G^F) \simeq G^F(C)$$

where the last isomorphism is by the Yoneda lemma.

Now the assignment $C \mapsto \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}(h_C \times F, G)$ defines a functor $\mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$, which we denote by G^F. At the same time, this construction defines a functor $(-)^F : \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}} \to \mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$, which is right adjoint to $(-) \times F$. It is a nice exercise to prove this.

h) A monoid is never cartesian closed unless it's trivial.

Exercise 98. Show that every Boolean algebra is cartesian closed.

Exercise 99. Show that CABool is not cartesian closed [use 2.3].

Exercise 100. Show that a complete partial order is cartesian closed if and only if it's a frame [see 4.4].

Exercise 101. Let Ω be a frame. By the preceding exercise, it is cartesian closed; denote by $x{\to}y$ the exponent in Ω. This exercise is meant to let you show that $\mathcal{C}_\Omega$ is cartesian closed.

a) Show that Ω also has greatest lower bounds $\bigwedge B$ for all subsets B.

b) Given objects (X, E_X) and (Y, E_Y), define their exponent $(Y, E_Y)^{(X, E_X)}$ as (Y^X, E) where Y^X is the set of all functions $X \to Y$ in Set, and

$$E(f) = \bigwedge\{E_X(x){\to}E_Y(f(x)) | x \in X\}$$

Show that this defines a right adjoint (at (Y, E_Y)) of $(-) \times (X, E_X)$.

Some useful facts:

- $\mathcal{C}$ is cartesian closed if and only if it has finite products, and for each pair of objects X, Y there is an object Y^X and an arrow $\varepsilon : Y^X \times X \to Y$ such that for every Z and map $Z \times X \xrightarrow{f} Y$ there is a unique $Z \xrightarrow{\tilde{f}} Y^X$ such that

$$\begin{tikzcd}
Z \times X \arrow[rr, "f"] \arrow[dr, "\tilde{f} \times \mathrm{id}_X"'] & & Y \\
& Y^X \times X \arrow[ur, "\varepsilon"'] &
\end{tikzcd}$$

  commutes.

- In a ccc, there are natural isomorphisms $1^X \simeq 1$; $(Y \times Z)^X \simeq Y^X \times Z^X$; $(Y^Z)^X \simeq Y^{Z \times X}$.

- If a ccc has coproducts, we have $X \times (Y + Z) \simeq (X \times Y) + (X \times Z)$ and $Y^{Z+X} \simeq Y^Z \times Y^X$.

Exercise 102. Prove these facts.

Recall that two maps $Z \times X \to Y$ and $Z \to Y^X$ which correspond to each other under the adjunction are calles each other's *transposes*.

Exercise 103. In a ccc, prove that the transpose of a composite $Z \xrightarrow{g} W \xrightarrow{f} Y^X$ is

$$Z \times X \xrightarrow{g \times \mathrm{id}_X} W \times X \xrightarrow{\tilde{f}} Y$$

if $\tilde{f}$ is the transpose of f.

Lemma 7.2 *In a ccc, given $f : X' \to X$ let $Y^f : Y^X \to Y^{X'}$ be the transpose of*

$$Y^X \times X' \xrightarrow{\mathrm{id} \times \mathrm{f}} Y^X \times X \xrightarrow{\varepsilon} Y$$

Then for each $f : X' \to X$ and $g : Y \to Y'$ the diagram

$$\begin{tikzcd}
Y^X \arrow[r, "g^X"] \arrow[d, "Y^f"'] & Y'^X \arrow[d, "Y'^f"] \\
Y^{X'} \arrow[r, "g^{X'}"'] & Y'^{X'}
\end{tikzcd}$$

commutes.

Proof. By the exercise, the transposes of both composites are the top and bottom composites of the following diagram:

$$\begin{tikzcd}
 & Y'^X \times X' \arrow[r, "\mathrm{id} \times \mathrm{f}"] & Y'^X \times X \arrow[dr, "\varepsilon"] & \\
Y^X \times X' \arrow[ur, "g^X \times \mathrm{id}"] \arrow[r, "\mathrm{id} \times \mathrm{f}"] \arrow[dr, "Y^f \times \mathrm{id}"'] & Y^X \times X \arrow[ur, "g^X \times \mathrm{id}"] \arrow[r, "\varepsilon"] & Y \arrow[r, "g"] & Y' \\
 & Y^{X'} \times X' \arrow[ur, "\varepsilon"] \arrow[r, "g^{X'} \times \mathrm{id}"'] & Y'^{X'} \times X' \arrow[ur, "\varepsilon"'] &
\end{tikzcd}$$

This diagram commutes because the right hand "squares" are naturality squares for ε, the lower left hand square commutes because both composites are the transpose of Y^f, and the upper left hand square commutes because both composites are $g^X \times f$. ∎

Proposition 7.3 *For every ccc $\mathcal{C}$ there is a functor $\mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$, assigning Y^X to (X, Y), and given $g : Y \to Y'$ and $f : X' \to X$, $g^f : Y^X \to {Y'}^{X'}$ is either of the composites in the lemma.*

Exercise 104. Prove the proposition.

7.2 Typed λ-calculus and cartesian closed categories

The λ-calculus is an extremely primitive formalism about functions. Basically, we can form functions (by λ-abstraction) and apply them to arguments; that's all. Here I treat briefly the *typed* λ-calculus.

We start with a set $\mathcal{S}$ of *type symbols* $S_1, S_2, \ldots$

Out of $\mathcal{S}$ we make the set of types as follows: every type symbol is a type, and if T_1 and T_2 are types then so is $(T_1 \Rightarrow T_2)$.

We have also *terms* of each type (we label the terms like $t{:}T$ to indicate that t is a term of type T):

- we may have constants $c{:}T$ of type T;
- for every type T we have a denumerable set of variables $x_1{:}T, x_2{:}T, \ldots$;
- given a term $t{:}(T_1 \Rightarrow T_2)$ and a term $s{:}T_1$, there is a term $(ts){:}T_2$;
- given $t{:}T_2$ and a variable $x{:}T_1$ there is a term $\lambda x.t{:}T_1 \Rightarrow T_2$.

Terms $\lambda x.t$ are said to be formed by *λ-abstraction*. This procedure binds the variable x in t. Furthermore we have the usual notion of substitution for free variables in a term t (types have to match, of course). Terms of form (ts) are said to be formed by *application*.

In the λ-calculus, the only statements we can make are equality statements about terms. Again, I formulate the rules in terms of theories. First, let us say that a *language* consists of a set of type symbols and a set of constants, each of a type generated by the set of type symbols.

An *equality judgement* is an expression of the form $\Gamma | t = s{:}T$ (to be read: "Γ thinks that s and t are equal terms of type T "), where Γ is a finite set of variables which includes all the variables free in either t or s, and t and s are terms of type T.

A *theory* is then a set $\mathcal{T}$ of equality judgements which is closed under the following rules:

i) $\Gamma | t = s{:}T$ in $\mathcal{T}$ implies $\Delta | t = s{:}T$ in $\mathcal{T}$ for every superset Δ of Γ;

ii) $FV(t) | t = t{:}T$ is in $\mathcal{T}$ for every term $t{:}T$ of the language (again, $FV(t)$ is the set of free variables of t);
if $\Gamma | t = s{:}T$ and $\Gamma | s = u{:}T$ are in $\mathcal{T}$ then so is $\Gamma | t = u{:}T$;

iii) if $t(x_1,\ldots,x_n){:}T$ is a term of the language, with free variables $x_1{:}S_1,\ldots,x_n{:}S_n$, and $\Gamma|s_1 = t_1{:}S_1,\ldots,\Gamma|s_n = t_n{:}S_n$ are in $\mathcal{T}$ then

$$\Gamma|t[s_1/x_1,\ldots,s_n/x_n] = t[t_1/x_1,\ldots,t_n/x_n]{:}T$$

is in $\mathcal{T}$;

iv) if t and s are terms of type $(T_1{\Rightarrow}T_2)$, x a variable of type T_1 which does not occur in t or s, and $\Gamma \cup \{x\}|(tx) = (sx){:}T_2$ is in $\mathcal{T}$, then $\Gamma \setminus \{x\}|t = s{:}(T_1{\Rightarrow}T_2)$ is in $\mathcal{T}$;

v) if $s{:}T_1$ and $t{:}T_2$ are terms and x a variable of type T_2, then

$$FV(s) \setminus \{x\} \cup FV(t)|((\lambda x.s)t) = s[t/x]{:}T_1$$

is in $\mathcal{T}$.

Given a language, an interpretation of it into a ccc $\mathcal{C}$ starts by choosing objects $[\![S]\!]$ of $\mathcal{C}$ for every type symbol S. This then generates objects $[\![T]\!]$ for every type T by the clause

$$[\![T_1{\Rightarrow}T_2]\!] = [\![T_2]\!]^{[\![T_1]\!]}$$

The interpretation is completed by choosing interpretations

$$1 \xrightarrow{[\![c]\!]} [\![T]\!]$$

for every constant $c{:}T$ of the language.

Such an interpretation then generates, in much the same way as in chapter 4, interpretations of all terms. For a finite set $\Gamma = \{x_1{:}T_1,\ldots,x_n{:}T_n\}$ let's again write $[\![\Gamma]\!]$ for the product $[\![T_1]\!] \times \cdots \times [\![T_n]\!]$ (this is only defined modulo a permutation of the factors of the product, but that will cause us no trouble).

The interpretation of $t{:}T$ will now be an arrow

$$[\![FV(t)]\!] \xrightarrow{[\![t]\!]} [\![T]\!]$$

defined as follows:

- $[\![x]\!]$ is the identity on $[\![T]\!]$ for every variable $x{:}T$;
- given $[\![t]\!] : [\![FV(t)]\!] \to [\![T_2]\!]^{[\![T_1]\!]}$ and $[\![s]\!] : [\![FV(s)]\!] \to [\![T_1]\!]$ we let $[\![(ts)]\!] : [\![FV((ts))]\!] \to [\![T_2]\!]$ be the composite

$$[\![FV((ts))]\!] \xrightarrow{\langle [\![t]\!]\pi_t, [\![s]\!]\pi_s \rangle} [\![T_2]\!]^{[\![T_1]\!]} \times [\![T_1]\!] \xrightarrow{\varepsilon} [\![T_2]\!]$$

  where π_t and π_s are the projections from $[\![FV((ts))]\!]$ to $[\![FV(t)]\!]$ and $[\![FV(s)]\!]$, respectively;

- given $[\![t]\!] : [\![FV(t)]\!] \to [\![T_2]\!]$ and the variable $x{:}T_1$ we let $[\![\lambda x.t]\!] : [\![FV(t) \setminus \{x\}]\!] \to [\![T_2]\!]^{[\![T_1]\!]}$ be the transpose of

$$[\![FV(t) \setminus \{x\}]\!] \times [\![T_1]\!] \xrightarrow{\tilde{t}} [\![T_2]\!]$$

where, if x occurs free in t so $[\![FV(t) \setminus \{x\}]\!] \times [\![T_1]\!] \simeq [\![FV(t)]\!]$, $\tilde{t}$ is just $[\![t]\!]$; and if x doesn't occur in t, $\tilde{t}$ is $[\![t]\!]$ composed with the obvious projection.

We now say that an equality judgement $\Gamma | t = s{:}T$ is *true* in this interpretation, if the diagram

$$\begin{tikzcd}
& {[\![FV(t)]\!]} \arrow[dr, "{[\![t]\!]}"] & \\
{[\![\Gamma]\!]} \arrow[ur, "\pi_t"] \arrow[dr, "\pi_s"'] & & {[\![T]\!]} \\
& {[\![FV(s)]\!]} \arrow[ur, "{[\![s]\!]}"'] &
\end{tikzcd}$$

commutes (again, π_s and π_t projections).

Lemma 7.4 *Let $t(x_1, \ldots, x_n){:}T$ have free variables $x_i{:}T_i$ and let $t_i{:}T_i$ be terms. Let*

$$\tilde{t_i} : [\![FV(t[t_1/x_1, \ldots, t_n/x_n])]\!] \to [\![T_i]\!]$$

be the obvious composite of projection and $[\![t_i]\!]$.

Then the composition

$$[\![FV(t[t_1/x_1, \ldots, t_n/x_n])]\!] \xrightarrow{\langle \tilde{t_i} | i=1 \ldots n \rangle} \prod_{i=1}^{n} [\![T_i]\!] = [\![FV(t)]\!] \xrightarrow{[\![t]\!]} [\![T]\!]$$

is the interpretation $[\![t[t_1/x_1, \ldots, t_n/x_n]]\!]$.

Exercise 105. Prove the lemma [take your time. This is not immediate].

Theorem 7.5 *Let S be a set of equality judgements and $\mathcal{T} = Cn(S)$ be the least theory containing S. If every judgement of S is true in the interpretation, so is every judgement in $\mathcal{T}$.*

Proof. Again, we show that the set of true judgements is a theory, i.e. closed under the rules in the definition of a theory.
i) and ii) are trivial;
iii) follows at once by lemma 7.4;

iv) Since $\Gamma \subseteq (\Gamma \setminus \{x\}) \cup \{x\}$ and because of the inductive hypothesis, we have that

$$\begin{tikzcd}
& [\![FV(s)]\!] \times [\![T_1]\!] \arrow[r, "{[\![s]\!] \times \mathrm{id}}"'] & [\![T_1 \Rightarrow T_2]\!] \times [\![T_1]\!] \arrow[dr, "\varepsilon"] & \\
[\![\Gamma \setminus \{x\}]\!] \arrow[ur, "\pi_s \times \mathrm{id}"] \arrow[dr, "\pi_t \times \mathrm{id}"'] & & & [\![T_2]\!] \\
& [\![FV(t)]\!] \times [\![T_1]\!] \arrow[r, "{[\![t]\!] \times \mathrm{id}}"'] & [\![T_1 \Rightarrow T_2]\!] \times [\![T_1]\!] \arrow[ur, "\varepsilon"'] &
\end{tikzcd}$$

commutes. Taking the transposes of both maps, we get the equality we want.
v) According to lemma 7.4, $[\![FV(s[t/x])]\!] \stackrel{[\![s[t/x]]\!]}{\longrightarrow} [\![T_1]\!]$ is

$$[\![FV(s[t/x])]\!] \stackrel{\tilde{t}}{\to} [\![FV(s)]\!] \stackrel{[\![s]\!]}{\to} [\![T_1]\!]$$

This is the same as

$$[\![FV(s[t/x])]\!] \stackrel{\langle \pi, [\![t]\!] \rangle}{\to} [\![FV(s) \setminus \{x\}]\!] \times [\![T_2]\!] \stackrel{[\![\lambda x.s]\!] \times \mathrm{id}}{\longrightarrow} [\![T_2 \Rightarrow T_1]\!] \times [\![T_2]\!] \stackrel{\varepsilon}{\to} [\![T_1]\!]$$

which is

$$[\![FV((\lambda x.s)t)]\!] \stackrel{[\![((\lambda x.s)t)]\!]}{\longrightarrow} [\![T_1]\!]$$

∎

There is also a *completeness theorem*: if a judgement $\Gamma | t = s{:}T$ is true in all possible interpretations, then every theory (in a language this judgement is in) contains it.

The relevant construction is that of a syntactic cartesian closed category out of a theory, and an interpretation into it which makes exactly true the judgements in the theory. The curious reader can find the, somewhat laborious, treatment in Lambek & Scott's "Higher Order Categorical Logic".

7.3 Representation of primitive recursive functions in ccc's with natural numbers object

Dedekind observed, that in Set, the set ω is characterized by the following property: given any set X, any element $x \in X$ and any function $X \stackrel{f}{\to} X$, there is a unique function $F : \omega \to X$ such that $F(0) = x$ and $F(x + 1) = f(F(x))$.

Lawvere took this up, and proposed this *categorical* property as a definition (in a more general context) of a "natural numbers object" in a category.

Definition 7.6 *In a category $\mathcal{C}$ with terminal object 1, a* natural numbers object *is a triple $(0, N, S)$ where N is an object of $\mathcal{C}$ and $1 \xrightarrow{0} N$, $N \xrightarrow{S} N$ arrows in $\mathcal{C}$, such that for any other such diagram*

$$1 \xrightarrow{x} X \xrightarrow{f} X$$

there is a unique *map $\phi : N \to X$ making*

$$\begin{tikzcd} 1 \arrow[r, "0"] \arrow[dr, "x"'] & N \arrow[r, "S"] \arrow[d, "\phi"] & N \arrow[d, "\phi"] \\ & X \arrow[r, "f"'] & X \end{tikzcd}$$

commute.

Of course we think of 0 as the zero element, and of S as the successor map. The defining property of a natural numbers object enables one to "do recursion", a weak version of which we show here: we show that every primitive recursive function can be represented in a ccc with natural numbers object.

Definition 7.7 *Let $\mathcal{C}$ be a ccc with natural numbers object $(0, N, S)$. We say that a number-theoretic function $F : \mathbb{N}^k \to \mathbb{N}$ is* represented *by an arrow $f : N^k \to N$ if for any k-tuple of natural numbers $n_1, \ldots n_k$, the diagram*

$$\begin{tikzcd} 1 \arrow[r, "0"] \arrow[dr, "0"'] & N \arrow[r, "\langle S^{n_1}, \ldots, S^{n_k} \rangle"] & N^k \arrow[d, "f"] \\ & N \arrow[r, "S^{F(n_1, \ldots, n_k)}"'] & N \end{tikzcd}$$

commutes.

Recall that the class of *primitive recursive* functions is given by the following clauses:

- The constant zero function $\lambda \vec{x}.0 : \mathbb{N}^k \to \mathbb{N}$, the function $\lambda x.x + 1 : \mathbb{N} \to \mathbb{N}$ and the projections $\lambda \vec{x}.x_i : \mathbb{N} \to \mathbb{N}$ are primitive recursive;
- The primitive recursive functions are closed under composition: if $F_1, \ldots, F_k : \mathbb{N}^l \to \mathbb{N}$ and $G : \mathbb{N}^k \to \mathbb{N}$ are primitive recursive, then so is $G(\langle F_1, \ldots, F_k \rangle) : \mathbb{N}^l \to \mathbb{N}$;
- The primitive recursive functions are closed under *definition by primitive recursion*: if $G : \mathbb{N}^k \to \mathbb{N}$ and $H : \mathbb{N}^{k+2} \to \mathbb{N}$ are primitive recursive, and $F : \mathbb{N}^{k+1} \to \mathbb{N}$ is defined by $F(0, \vec{x}) = G(\vec{x})$ and $F(n + 1, \vec{x}) = H(n, F(n, \vec{x}), \vec{x})$ then F is primitive recursive.

Proposition 7.8 *In a ccc $\mathcal{C}$ with natural numbers object, every primitive recursive function is representable.*

Proof. I do only the case for definition by primitive recursion. So by inductive hypothesis we have arrows G and H representing the homonymous functions. By interpretation of the λ-calculus, I use λ-terms: so there is an arrow

$$\lambda\vec{x}.G(\vec{x}) : 1 \to N^{(N^k)}$$

and an arrow

$$\lambda\vec{x}.H(n,\phi(\vec{x}),\vec{x}) : N^{(N^k)} \times N \to N^{(N^k)}$$

which is the interpretation of a term with free variables $\phi{:}N^{(N^k)}$ and $n{:}N$; this map is the exponential transpose of the map which intuitively sends $(n,\phi,\vec{x})$ to $(n,\phi(\vec{x}),\vec{x})$. Now look at

$$1 \xrightarrow{\langle\lambda\vec{x}.G(\vec{x}),0\rangle} N^{(N^k)} \times N \xrightarrow{(\lambda\vec{x}.H(n,\phi(\vec{x}),\vec{x}))\times S} N^{(N^k)} \times N$$

By the natural numbers object property, there is now a unique map

$$\bar{F} = \langle\tilde{F},\sigma\rangle : N \to N^{(N^k)} \times N$$

which makes the following diagram commute:

$$\begin{tikzcd}
1 \arrow[r, "0"] \arrow[dr, "\langle\lambda\vec{x}.G(\vec{x}),0\rangle"'] & N \arrow[r, "S"] \arrow[d, "\bar{F}"] & N \arrow[d, "\bar{F}"] \\
 & N^{(N^k)} \times N \arrow[r, "(\lambda\vec{x}.H(n,\phi(\vec{x}),\vec{x}))\times S"'] & N^{(N^k)} \times N
\end{tikzcd}$$

One verifies that σ is the identity, and that the composite

$$N^{k+1} \xrightarrow{\tilde{F}\times\mathrm{id}} N^{(N^k)} \times N^k \xrightarrow{\varepsilon} N$$

represents F. ∎

Exercise 106. Make these verifications.

That's it!!

Index