An introduction to fibrations, topos theory, the effective topos and modest sets

Wesley Phoa
wkp@dcs.ed.ac.uk

Abstract

A *topos* is a categorical model of constructive set theory. In particular, the *effective topos* is the categorical 'universe' of recursive mathematics. Among its objects are the *modest sets*, which form a set-theoretic model for polymorphism. More precisely, there is a *fibration* of modest sets which satisfies suitable categorical completeness properties, that make it a model for various polymorphic type theories.

These lecture notes provide a reasonably thorough introduction to this body of material, aimed at theoretical computer scientists rather than topos theorists. Chapter 2 is an outline of the theory of fibrations, and sketches how they can be used to model various typed λ-calculi. Chapter 3 is an exposition of some basic topos theory, and explains why a topos can be regarded as a model of set theory. Chapter 4 discusses the classical PER model for polymorphism, and shows how it 'lives inside' a particular topos—the effective topos—as the category of modest sets. An appendix contains a full presentation of the internal language of a topos, and a map of the effective topos.

Chapters 2 and 3 provide a sampler of categorical type theory and categorical logic, and should be of more general interest than Chapter 4. They can be read more or less independently of each other; a connection is made at the end of Chapter 3.

The main prerequisite for reading these notes is some basic category theory: limits and colimits, functors and natural transformations, adjoints, cartesian closed categories. No knowledge of indexed categories or categorical logic is needed. Some familiarity with 'ordinary' logic and typed λ-calculus is assumed.

Contents

Chapter 1

Introduction

The simply typed λ-calculus can be modelled in a very naive way, using sets and functions: arrow types are interpreted as ordinary set-theoretic function spaces. When John Reynolds first introduced the polymorphic λ-calculus into computer science, he hoped that it would have a straighforward set-theoretic semantics too. Unfortunately, he later showed that this was impossible: [19].

A few years later, Andy Pitts showed that the second order λ-calculus ("System F") *did* have set-theoretic models—provided one worked, not in ordinary classical set theory, but in *constructive* set theory: [17]. Category theory plays a central role in the argument.

There is a natural categorical semantics for the simply typed λ-calculus: namely, using cartesian closed categories. Scott observed that any such model could be regarded as 'set-theoretic' by embedding it, via the Yoneda functor, in a *topos:* a categorical model of constructive set theory. In essence, Pitts simply extended this technique to models of System F.

Though very interesting, such results are perhaps not entirely natural: though the models are technically 'set-theoretic', they were really only *forced* to be set-theoretic using the Yoneda embedding. Whether topos-theoretic models of polymorphism arose in a more direct way remained an open question.

Models for System F, and more complicated polymorphic type theories, had been known for a long time. In particular, Girard had already discovered the so-called PER model in his original work on System F. A few years ago Moggi noticed, and Hyland proved, that the PER model actually 'lives inside' Hyland's effective topos $\mathcal{E}\!f\!f$, and thus provides quite a natural set-theoretic model of polymorphism: PERs correspond to particular objects ('sets') in $\mathcal{E}\!f\!f$, what Scott dubbed the *modest sets*.

This provides a more convincing solution to Reynolds' problem: the modest sets have a nice logical characterisation and better completeness properties. In fact it turns out that one can model, not just System F, but various other systems all the way up to the calculus of constructions of Coquand and Huet. This sets earlier work of Longo and Moggi on 'ω-set' interpretations in a topos-theoretic context.

Unfortunately, these results of Hyland's—refined, elaborated and corrected by Hyland, Robinson and Rosolini—remain rather inaccessible to the people who would find them most useful, namely computer scientists. A rather intimidating amount of category theory and topos theory is required to understand them, and the original papers mostly assume that the reader has this background.

These notes give a very *ahistorical* account of this background knowledge, and of the results of Hyland *et al.* They are aimed at a computer scientist who is reasonably familiar with category theory, or a graduate student who has done a first course in category theory. I think that, although there is a great deal of material to cover, the effort is worth it: the insights one gains are very powerful (for example, they make possible a much smoother and simpler development of 'synthetic domain theory').

Furthermore, the bits of knowledge one picks up along the way, such as categorical logic and the categorical approach to type theory, are important in themselves. A striking example is Lawvere's insight that the quantifiers can be regarded as adjoints to substitution.

I've left out most of the long technical arguments, though I have tried to include enough proofs—and exercises—to give the flavour of the material. The goal is to give an 'insider's view' of the material without overwhelming the reader. Full details may be found in the references, to which these notes are only a stepping-stone.

All this forms part of a projected series of much more comprehensive lecture notes. At the moment, most of these notes only have a kind of fantasy existence in my head. A full version ought to include an account of (a) j-operators, (b) realizability toposes in general, (c) modest sets as (co)discrete objects, and (d) domains in realizability toposes. It would also contain a fuller bibliography; the present one only contains references explicitly cited in the text.

There are other, less categorical, approaches to presenting all of this material; for example, Rosolini and Scott have given a presentation within the framework of intuitionistic ZF. From the point of view of applications, there is much to be said for this tack. However, I think that many of the results are best understood categorically, at least if one really does want an insider's view of the results.

Fibrations

Thanks to Seely ([22], [23]) and others, and later to Hyland and Pitts and more others, we know that all kinds of typed λ-calculi can be interpreted using category theory. The usual way is to use *indexed categories*. It turns out to be technically easier to use *fibrations*, which are basically[1] the same thing. That's what chapter 2 is about: it's an exposition of the theory of fibrations, with an eye to type theory.

Those notes were written to accompany the informal talks I gave in the categories club at the LFCS. A detailed account of fibrations, and their relationship to type theory, appears in Bart Jacobs' thesis [11]; I think [3] and [10] should be useful too. They cover many important topics which are not mentioned here—in paticular, the relationship between fibrations and internal categories. Also, results stated without proof are proved in Bart's thesis. Or at least, the thesis contains a reference to a proof.

Readers who already know about indexed categories may wonder why they should bother about fibrations. The usual answer is that fibrations sometimes arise more naturally, that they have a number of technical advantages and that that they are foundationally less problematic. But for the purposes of these notes, the difference is really just a matter of taste. Propaganda aside, indexed categories and (cloven) fibrations are really equivalent notions, and informally one tends to think of them interchangeably. In particular, I don't believe fibrations are really 'harder' then indexed categories; so, since they do turn out to be more convenient later on, we might as well use them from the start.

Incidentally, no prior knowledge of indexed categories is assumed.

This chapter is *not* a systematic introduction to fibrations or indexed categories, or to categorical type theory. And it omits the treatment of dependent type theory. This is dealt with in

[1] but not really

the following chapter—not in full generality, but only for toposes.

Also, Chapters 2 and 3 are meant to be more or less *independent* (and, as a consequence, there are some redundancies built in). However, the material in Chapter 2 is needed in the final section of Chapter 3.

Toposes

Chapter 3 is a brief outline of some basic topos theory; it's only meant to explain what a topos is and why one can think of a topos as a universe of sets and reason about it using set-like notation. I haven't tried to give a completely systematic development. The best reference is Johnstone's book, which parts of these notes follow very closely. For the internal language of a topos, it's better to read Lambek/Scott or Bell. Goldblatt provides a gentle introduction, and the first few chapters are recommended. There are lots of other references.

In a way this account was written to complement books like Lambek/Scott and Bell. Those books have quite a 'logical' flavour, while the present notes, like Johnstone's book, are more 'categorical'. I think it's important to appreciate both points of view.

Many of these exercises (but not the incorrect ones) are taken from Johnstone, and you may find the solutions there. None of them are meant to be hard (except the incorrect ones) and most of them should be quite short.

There are many different ways to present the internal language; the version we use is a very rich one. We'll give an 'internal' semantics for it. If you are familiar with forcing, you may find the Kripke-Joyal semantics more intuitive. Fourman's treatment, based on the Fourman/Scott theory of identity and existence, is convenient for reasoning about possibly undefined elements of sets. It's good to see different points of view.

There's a lot more to topos theory than these notes could possibly convey, and many of the topics not discussed here are actually quite relevant to computing: for example, the Hyland/Pitts model of the theory of constructions. It is very important to realise that topos theory has many sides and the view presented here—of a topos as a model of constructive set theory—is only one of them. I hope that some readers will be encouraged to learn more about this beautiful subject.

For ease of reference, the full internal language is set out in the *appendix*.

Aside. As an interpretation of constructive logic, topos theory models *provability*, not *proofs*; this distinguishes it from (categorical models of) the theory of constructions. In fact the notion of explicit proof-object, familiar from the work of Martin-Löf, has a categorical counterpart; this idea was developed independently by Lawvere in the 60s, but the connection was only made precise by Seely in [22]. More recently, Pavlovič has been attempting to generalise topos theory along these lines to incorporate the idea of proof-relevance. The results look promising, but the theory is not in a settled state.

The effective topos

Chapter 4 provides an informal introduction to the effective topos, aimed mainly at computer scientists. I have tried to motivate its construction by discussing the PER model of polymorphism, and the category of ω-sets (or 'assemblies'). However, it's worth remembering that the effective topos actually predates all this, and really grew out of earlier work on realizability, recursive mathematics, generalised recursion theory and models of constructive logic. Hyland's original paper represents this point of view.

In fact the prehistory of the effective topos is complicated, and the situation in the '60s and '70s remains quite obscure to me. I will not try to give a coherent account. Even if I knew all the details, they would probably be distracting. It seemed better to make up a bogus history which (I like to think) makes more sense than the truth.

Incidentally, the description of $\mathcal{E}\!f\!f$ given here is Hyland's original one, which seems to be quite closely related to Powell's. There are several alternative ways to construct it, due to Carboni/Freyd/Scedrov, Robinson/Rosolini and others. Perhaps the most illuminating is one proposed by Colin McLarty, in a paper which will hopefully appear soon. It is the Powell/Hyland and McLarty constructions which give the best insight into the logic of $\mathcal{E}\!f\!f$, I'd say.

There is much more to tell than I could relate here. Sadly, it's not enough just to refer the reader to the references: many of the interesting logical applications of realizability toposes have never appeared in print. Perhaps one day a more definitive account will be available. The present author[2] has not enough experience in logic or metamathematics to write one.

Quite a lot of different categories and fibrations are defined and related to each other. The map in the appendix may help clarify the situation. Note, however, that it doesn't say *everything*.

Warning. The effective topos is a special case of a more general construction: any partial combinatory algebra (for example, any model of the untyped λ-calculus) gives rise to a corresponding *realizability topos*. Because so many notions of computation—those arising from partial recursive functions, from λ-expressions, and so on—were proved to be 'equivalent', one might expect the associated realizability toposes to be equivalent: but[3] *they are not.*[4] So it is misleading to speak of a single 'realizability universe'.

Acknowledgements In revising the notes on fibrations, I have benefited from the advice of many members of the Categories Club at the LFCS, who attended some talks of mine organised by Barry Jay. The notes on toposes and the effective topos were written for an informal graduate course at the LFCS, organised by Stuart Anderson.

The version of the internal language of a topos presented here is original to these notes: it contains many more features than usual (e.g. subset types, coproducts, dependent types) and is thus adapted to the purposes of chapter 4. I also believe it is a convenient language to work with generally[5]. It is partly based on conversations with James McKinna; Healfdene Goguen helped with some of the rules.

The notes on the effective topos grew out of an expository report on the effective topos and modest sets which I wrote as an undergraduate in 1987, and I should thank my honours supervisor Martin Ward, my category theory lecturer Tim Brook and my colleague Scott Johnson for their support in this effort. My account of toposes was greatly influenced by Tim Brook's lectures and by Peter Johnstone's book.

---

[2] viz. me

[3] as pointed out by Johnstone and Robinson

[4] An easy example: the partial recursive functions and the r.e. graph model both give rise to the same class of maps between natural numbers—viz. the recursive functions—and one might naively hope that they give rise to the 'same' model of set theory; but the functionals of higher type in each model satisfy quite different continuity properties—for example, all type 2 functionals over the r.e. graph model are uniformly continuous, and this fails for the effective operations—so they can't be the same at all. As a matter of fact, realizability toposes over term models of the λ-calculus probably do not even satisfy Church's Thesis, despite the fact that 'externally' all the functions on the natural numbers are recursive.

[5] There are tentative plans to implement a proof assistant based on this language.

Chapter 2

Fibrations

2.1 Motivation

Suppose we are interested in modelling the simply typed λ-calculus, without type variables. For simplicity we assume that there is just one ground type ι, so the types are

$$\tau \quad ::= \quad \iota \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2$$

We can interpret them in any cartesian closed category $\mathbb{C}$: just pick an object $N \in \mathbb{C}$ and put

$$[\![\iota]\!] = N, \quad [\![\tau_1 \times \tau_2]\!] = [\![\tau_1]\!] \times [\![\tau_2]\!], \quad [\![\tau_1 \to \tau_2]\!] = [\![\tau_2]\!]^{[\![\tau_1]\!]}$$

Then the interpretation of a term $t : \tau\,[x : \sigma]$ is a map $[\![t]\!]\colon [\![\sigma]\!] \to [\![\tau]\!]$.

For example, we could use $\mathcal{S}ets$, let ι denote the natural numbers and interpret product and arrow types by cartesian products and set-theoretic function spaces.

Now, what if we want to be able to model variable types? That is, if we want to make sense of 'variable' type expressions like

$$\tau(X)\,[X : \mathsf{K}]$$

(where K is perhaps some 'kind' whose elements are types). We could interpret this using a cartesian closed category $\mathbb{C}$ by letting K denote some set $K \in \mathcal{S}ets$, and specifying a K-indexed family of objects of $\mathbb{C}$:

$$[\![\tau(X)]\!] = \{T_k\}_{k \in K} \quad \text{where } T_k \in \mathbb{C}_0$$

Here $\mathbb{C}_0$ denotes the set of objects of $\mathbb{C}$.

This is a pretty intuitive idea, but it is not always general enough. For example,

1. We might want to let K denote, not a set, but an object K in some other category $\mathbb{B}$. But then it makes no sense to talk about a K-indexed family of objects of $\mathbb{C}$.

2. We might want to interpret universally quantified types such as

   $$\Pi X : \mathsf{K}.\, T(X)$$

   as products

   $$\prod_{k \in K} T_k \in \mathbb{C}$$

   But these products may not exist in $\mathbb{C}$. For example, if K represents the collection of *all* types (so that K should be the set $\mathbb{C}_0$ of objects of $\mathbb{C}$) then it will be very hard to find a category $\mathbb{C}$ which has such products.

So here is the problem: given some 'base' category $\mathbb{B}$, whose objects are though of as 'indexing sets', what's a categorical way of talking about 'families of objects of a category, indexed by objects $I \in \mathbb{B}$'? In different words, we want a notion of 'category indexed over $\mathbb{B}$', or 'category relative to $\mathbb{B}$'. (Furthermore, we want to talk about I-indexed products and coproducts, and so on...)

There are two ways to define such things. One is say that a category over $\mathbb{B}$ is an *indexed category*: that is, a (pseudo)functor $F: \mathbb{B}^{op} \to \mathcal{C}at$ that takes $I \in \mathbb{B}$ to a category $F(I)$ which we regard as begin a category of 'I-indexed families of objects'. The other way is to think all the indexed families as living inside a single category $\mathbb{E}$, with a functor $p: \mathbb{E} \to \mathbb{B}$ taking a family to the object that indexes it: so a *fibration* is a structure $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ satisfying certain axioms.

These two approaches are very closely related: in fact, pseudofunctors $\mathbb{B}^{op} \to \mathcal{C}at$ (indexed categories) are basically the same thing as cloven fibrations, and real functors $\mathbb{B}^{op} \to \mathcal{C}at$ are the same thing as split fibrations. Indexed categories are more intuitive at first, while fibrations are technically more convenient in some ways[1].

2.2 Fibrations

Given a functor $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$, when does it make sense to regard each object $X \in \mathbb{E}$ as a family '$(X_i)_{i \in I}$' indexed by the object $I = pX \in \mathbb{B}$? To try to motivate the axioms, here is the simplest example.

Let $\mathbb{B} = \mathcal{S}ets$ and $\mathbb{E} = \mathrm{Fam}(\mathcal{S}ets)$ where $\mathrm{Fam}(\mathcal{S}ets)$ is defined as follows:

objects: families $\{X_i\}_{i \in I}$ where $I \in \mathcal{S}ets$ and $X_i \in \mathcal{S}ets$

arrows: $f: \{X_i\}_{i \in I} \to \{Y_j\}_{j \in J}$ is given by a function $\phi: I \to J$ (a reindexing function) together with a family of functions $\{f_i: X_i \to Y_{\phi(i)}\}$

There is an obvious functor $p: \mathrm{Fam}(\mathcal{S}ets) \to \mathcal{S}ets$ that takes $\{X_i\}_{i \in I} \mapsto I$. Let's look at some properties that arrows in $\mathrm{Fam}(\mathcal{S}ets)$ can have, in terms of this functor p.

Call a map $f: \{X_i\}_{i \in I} \to \{Y_j\}_{j \in J}$ *vertical* if $pf = 1_I$. That is, a vertical map over I is an I-indexed family of functions between two I-indexed families of sets: there is no reindexing going on. If we let $\mathbb{E}_I$ (also written $p^{-1}(I)$) be the subcategory of $\mathbb{E}$ consisting of the objects and vertical maps over I, then $\mathbb{E}_I$ is just $\mathcal{S}ets^I$, the category of I-indexed families of sets.

Call a map $f: \{X_i\}_{i \in I} \to \{Y_j\}_{j \in J}$ *cartesian* if each function f_i is an isomorphism. That is, all f does is reindex the family; none of the f_i 'do any work', and f is essentially determined by ϕ. Note that this definition is not purely categorical: it doesn't make sense for an arbitrary functor p.

It turns out that the key to defining fibrations is to characterise cartesian maps categorically. (Cartesian maps also correspond to the operation of substitution in type theory and logic.) So let's see what kinds of categorical properties the cartesian maps have.

Firstly, note that if we have a family $\{X_i\}_{i \in I}$, then any map $\alpha: K \to I$ in $\mathcal{S}ets$ has a canonical lifting to a cartesian map $f: \{X_{\alpha(k)}\}_{k \in K} \to \{X_i\}_{i \in I}$ in $\mathrm{Fam}(\mathcal{S}ets)$ (where each of the f_i's is just the identity map 1_{X_i}). In a picture,

[1]Fibrations are also foundationally more attractive—they are the models for a first-order, 'essentially algebraic' theory—and, unlike indexed categories, they make sense in a general 2-categorical setting. But these issues are not important to us here.

$$\begin{tikzcd}
\{X_{\alpha(k)}\}_{k\in K} \arrow[rr, dashed, "\exists\ \text{cartesian}"] & {} \arrow[d, dotted, "p"] & \{X_i\}_{i\in I} \\
K \arrow[rr, "\alpha"'] & {} & I
\end{tikzcd}$$

What's more, you can think of any cartesian map as being a map that arises like this: any cartesian map factors through one of this form, via a unique vertical isomorphism.

Secondly, note that cartesian maps have a certain universal property: an arbitrary map f factors through a cartesian map g precisely when $\phi = pf$ factors through $\alpha = pg$, and the factorization is uniquely determined. For a cartesian map g of the 'canonical' form just described, the picture looks as follows:

$$\begin{tikzcd}
\{Y_j\}_{j\in J} \arrow[dr, dashed, "\{f_j : Y_j \to X_{\psi(j)}\}"'] \arrow[drr, "\{f_j : Y_j \to X_{\phi(j)}\}"] & & \\
& \{X_{\alpha(k)}\}_{k\in K} \arrow[r] \arrow[d, dotted, "p"] & \{X_i\}_{i\in I} \\
J \arrow[dr, "\psi"'] \arrow[drr, "\phi"] & {} & \\
& K \arrow[r, "\alpha"'] & I
\end{tikzcd}$$

Now we can finally make some general definitions.

Let $\mathbb{B}$ and $\mathbb{E}$ be arbitrary categories and let $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ be a functor.

Definition 2.2.1 *A map $g: X' \to X$ in $\mathbb{E}$ is* ***cartesian*** *if, given any map $f: Y \to X$, each factorization of $\phi = pf$ through $\alpha = pg$ uniquely determines a factorization of f through g:*

$$\begin{tikzcd}
Y \arrow[dr, dashed, "\exists !\, f'"'] \arrow[drr, "f"] & & \\
& X' \arrow[r, "g"'] \arrow[d, dotted, "p"] & X \\
J \arrow[dr, "\psi"'] \arrow[drr, "\phi"] & {} & \\
& K \arrow[r, "\alpha"'] & I
\end{tikzcd}$$

We want a 'fibration' $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ to have the property that we can think of $\mathbb{E}$ as a category of families of things, indexed by objects in $\mathbb{B}$. In the definition of cartesian map, then, we think of X' as being the result of reindexing the family X along the map α. For all this to make sense, there need to be enough cartesian maps: we should always be able to reindex families when we expect to be able to.

Definition 2.2.2 $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *is a* ***fibration*** *if, for all $X \in \mathbb{E}$ and maps $\alpha: K \to I = pX$, there exists an object $X' \in \mathbb{E}$ and a cartesian map $g: X' \to X$ such that $pg = \alpha$:*

$$\begin{tikzcd} X' \arrow[rr, dashed, "\exists\ cartesian\ lifting\ g"] & \arrow[d, dotted, "p"] & X \\ K \arrow[rr, "\alpha"'] & {} & I \end{tikzcd}$$

We refer to X' as a ***reindexing of X along*** *α.*

This is a very general definition, and there are lots of uninteresting fibrations.

It's easy to show that cartesian liftings are 'unique up to isomorphism': if $g: X' \to X$ and $h: X'' \to X$ are two cartesian maps with $pg = ph$, then there is a unique isomorphism $k: X' \to X''$ such that $g = hk$. So the following definition seems innocuous.

Definition 2.2.3 *If* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *is a fibration, a* ***cleavage*** *for* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *is a particular choice of cartesian lifting $\alpha^*X \to X$ for each X, α. A fibration equipped with a* particular *cleavage is called a* ***cloven fibration****.*

Note that, because of the axiom of choice (in the *metatheory*—not 'in $\mathbb{B}$', whatever that might mean) any fibration has a cleavage. And in practice, we usually show that something is a fibration by exhibiting a particular cleavage. However, any given cleavage need not be especially well-behaved, and it is usually a bad idea to make general definitions that depend on a choice of cleavage. We'll say a little more about this later.

By the way, the definition of **vertical** map in $\mathbb{E}$ still makes sense, and we can talk about the category $\mathbb{E}_I = p^{-1}(I)$ of objects and vertical maps over some $I \in \mathbb{B}$. This is usually called the **fibre over** I, and we think of it as the category of I-indexed families.

It's often the case that all the fibres $\mathbb{E}_I$ have (say) binary products. The following definition is then useful. Note that it does not depend on a choice of cleavage.

Definition 2.2.4 *Let* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *be a fibration. We say that* ***cartesian lifting preserved binary products*** *if, whenever*

- *$X, Y \in \mathbb{E}_I$ and $\alpha: K \to I$;*
- *$X \times Y$ exists in $\mathbb{E}_I$; and*

- *$X', Y', Z' \in \mathbb{E}_K$ are reindexings of $X, Y, X \times Y$ along α*

then Z' is a product of X' and Y'.

Note that since any two reindexings of an object along α are isomorphic, this definition does not depend on the choice of reindexings. Also note that it is a little imprecise, since it does not require that the product projections $X' \leftarrow Z' \rightarrow Y'$ be 'reindexings' of the original ones.

Exercise 2.2.5 Make the above definition precise, by first showing that there are natural candidates for the product projections $X' \leftarrow Z' \rightarrow Y'$. (Use the universal property of cartesian maps to construct them.) ■

If $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ is cloven, the condition is equivalent to a more natural-looking one: that reindexing functors preserve binary products (for the definition of 'reindexing functor', see the remarks after definition 2.2.14). Also, note that we can define what it means for cartesian lifting to preserve finite limits, exponentials etc. in a completely analogous way.

Definition 2.2.6 *A fibration $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ is a* ***fibred CCC*** *if each fibre $\mathbb{E}_I$ is a cartesian closed category, and cartesian lifting preserves binary products and exponentials.*

Example 2.2.7 This is a mild generalisation of the first example. For any category $\mathbb{C}$, we will construct a fibration $\mathrm{Fam}(\mathbb{C})$ over $\mathcal{S}ets$. The category $\mathrm{Fam}(\mathbb{C})$ is defined as follows:

objects: families $\{X_i\}_{i \in I}$ where $I \in \mathcal{S}ets$ and $X_i \in \mathbb{C}$

arrows: $f: \{X_i\}_{i\in I} \rightarrow \{Y_j\}_{j \in J}$ is given by a function $\phi: I \rightarrow J$ in $\mathcal{S}ets$
together with a family of maps $\{f_i: X_i \rightarrow Y_{\phi(i)}\}$ in $\mathbb{C}$

Again there is an obvious functor $p: \mathrm{Fam}(\mathbb{C}) \rightarrow \mathcal{S}ets$, and, for the same reasons as before, we have a fibration. It is a fibred CCC if and only if $\mathbb{C}$ is cartesian closed. ■

When we are working with a category $\mathbb{C}$, and talking about families of objects of $\mathbb{C}$ and so on, then in a sense we are really working with the fibration $\begin{smallmatrix}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{smallmatrix}$. So the theory of fibrations can be viewed as a generalisation of category theory, in which $\mathcal{S}ets$ is replaced by some other category $\mathbb{B}$.

Example 2.2.8 Let $\mathbb{C}$ be the full subcategory of $\mathcal{S}ets$ whose objects are subsets of $\mathbb{N}$, and in which a map $X \rightarrow Y$ is given by a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that $f(X) \subseteq Y$. There are two different fibrations $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathcal{S}ets\end{smallmatrix}$ which we can regard as 'indexed' versions of $\mathbb{C}$:

1. The 'standard' one, where $\mathbb{E}$ is the following category:

   objects: families $\{X_i\}_{i \in I}$ where $I \in \mathcal{S}ets$ and $X_i \subseteq \mathbb{N}$

   arrows: $f: \{X_i\}_{i\in I} \rightarrow \{Y_j\}_{j \in J}$ is given by a function $\phi: I \rightarrow J$ in $\mathcal{S}ets$
   together with a family of functions $\{f_i: \mathbb{N} \rightarrow \mathbb{N}\}_{i \in I}$
   such that $f_i(X_i) \subseteq Y_{\phi(i)}$ for all $i \in I$

2. The 'uniform' one: $\mathbb{E}$ has the same objects, but a map $f: \{X_i\}_{i \in I} \to \{Y_j\}_{j \in J}$ is given by a function $\phi: I \to J$ and a *single* function $f: \mathbb{N} \to \mathbb{N}$ such that $f(X_i) \subseteq Y_{\phi(i)}$ for all $i \in I$.

It is important to understand the difference between these two fibrations. This particular example is not so important, but the same kind of thing comes up when we define fibrational PER models of polymorphism: to get the completeness we need to model polymorphic types, the definition of maps has to be 'uniform' rather than 'standard'. ■

Example 2.2.9 Let $\mathbb{B}$ be a category with pullbacks, and let $\mathbb{E}$ be the arrow category $\mathbb{B}^{\to}$ defined as follows:

objects: arrows $\left(\begin{smallmatrix} J \\ \downarrow\phi \\ I \end{smallmatrix}\right)$ in $\mathbb{B}$

arrows: a morphism $f: \left(\begin{smallmatrix} J \\ \downarrow\phi \\ I \end{smallmatrix}\right) \to \left(\begin{smallmatrix} J' \\ \downarrow\phi' \\ I' \end{smallmatrix}\right)$ is given by a commutative square

$$\begin{tikzcd} J \arrow[r, "\beta"] \arrow[d, "\phi"'] & J' \arrow[d, "\phi'"] \\ I \arrow[r, "\alpha"'] & I' \end{tikzcd}$$

The functor $\begin{smallmatrix} \mathbb{B}^{\to} \\ \downarrow \\ \mathbb{B} \end{smallmatrix}$ is the codomain functor $cod: \mathbb{B}^{\to} \to \mathbb{B}$ that maps

$$\left(\begin{smallmatrix} J \\ \downarrow \\ I \end{smallmatrix}\right) \mapsto I, \qquad f \mapsto \alpha$$

A map $f: \left(\begin{smallmatrix} J \\ \downarrow \\ I \end{smallmatrix}\right) \to \left(\begin{smallmatrix} J' \\ \downarrow \\ I' \end{smallmatrix}\right)$ is cartesian if and only if it's a pullback square (check this); and given $\left(\begin{smallmatrix} J \\ \downarrow\phi \\ I \end{smallmatrix}\right) \in \mathbb{B}^{\to}$ and $\alpha: K \to I$ in $\mathbb{B}$, we can obtain a cartesian lifting of α by forming a pullback

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & J \arrow[d, "\phi"] \\ K \arrow[r, "\alpha"'] & I \end{tikzcd}$$

So a cleavage for this fibration is just a particular choice of pullback, for each pair of maps with the same codomain. Also, note that the fibre $p^{-1}(I)$ is just the slice category $\mathbb{B}/I$, whose objects are maps $\left(\begin{smallmatrix} X \\ \downarrow \\ I \end{smallmatrix}\right)$ in $\mathbb{B}$, and whose arrows are commutative triangles: see definition 3.4.1. ■

This last example has an intuitive meaning: it presents $\mathbb{B}$ as a category 'indexed over itself'. In fact, in the case $\mathbb{B} = \mathcal{S}ets$, it is 'the same' as Fam($\mathcal{S}ets$). We need another definition before we can make this precise.

Definition 2.2.10 *Let* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *and* $\begin{smallmatrix}\mathbb{F}\\ \downarrow q\\ \mathbb{B}\end{smallmatrix}$ *be fibrations. A* ***cartesian functor*** f *from* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *to* $\begin{smallmatrix}\mathbb{F}\\ \downarrow q\\ \mathbb{B}\end{smallmatrix}$ *is a functor* $f\colon \mathbb{E} \to \mathbb{F}$ *such that*

- f *takes cartesian maps to cartesian maps; and*
- *The triangle*

$$\begin{tikzcd} \mathbb{E} \arrow[rr, "f"] \arrow[dr, "p"'] & & \mathbb{F} \arrow[dl, "q"] \\ & \mathbb{B} & \end{tikzcd}$$

  commutes.

If f *and* g *are cartesian functors from* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *to* $\begin{smallmatrix}\mathbb{F}\\ \downarrow q\\ \mathbb{B}\end{smallmatrix}$, *a* ***cartesian natural transformation*** $f \Rightarrow g$ *is a natural transformation* $\mu\colon f \Rightarrow g$ *whose components are all vertical maps.*

Fibrations over a given base category $\mathbb{B}$, cartesian functors and cartesian natural transformations form a 2-category; so it makes sense to talk about cartesian ('fibred') adjunctions, and so on.

Example 2.2.11 There is a natural correspondence between maps $\left(\begin{smallmatrix}X\\ \downarrow\\ I\end{smallmatrix}\right)$ in $\mathcal{S}ets$ and I-indexed families $\{X_i\}_{i\in I}$ of sets:

$$\begin{array}{ccll} \{X_i\}_{i\in I} & \mapsto & \left(\begin{smallmatrix}J\\ \downarrow\phi\\ I\end{smallmatrix}\right) & \text{where } J = \coprod_{i\in I} X_i \text{ and } \phi(X_i) = i \\ \left(\begin{smallmatrix}J\\ \downarrow\phi\\ I\end{smallmatrix}\right) & \mapsto & \{X_i\}_{i\in I} & \text{where } X_i = \phi^{-1}\{i\} \end{array}$$

Both of these constructions define functors which preserve cartesian maps, and they define an equivalence (of fibrations) between the two fibrations over $\mathcal{S}ets$: $\mathrm{Fam}(\mathcal{S}ets) \simeq \mathcal{S}ets^{\to}$. ■

Note that cartesian functors are not required to preserve cleavages, if they exist (just as left exact functors are not required to preserve *designated* limit cones in categories with finite limits).

We could have required that the triangle in the definition of cartesian functor only commute up to natural isomorphism. This would be a more natural definition from a mathematical point of view, but it complicates things. In the examples that follow the triangle turns out to commute on the nose—so we can stick with the definition we've got[2].

[2]If one really wants to take these 2-categorical issues seriously, one needs to use a more sophisticated definition of 'fibration'.

Definition 2.2.12 *Let* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *be a fibration and let* $f\colon \mathbb{B}' \to \mathbb{B}$ *be any functor. Form the 2-pullback*

$$\begin{tikzcd}
\mathbb{E}' \arrow[r] \arrow[d, "p'"'] & \mathbb{E} \arrow[d, "p"] \\
\mathbb{B}' \arrow[r, "f"'] & \mathbb{B}
\end{tikzcd}$$

Then $\begin{smallmatrix}\mathbb{E}'\\ \downarrow p'\\ \mathbb{B}'\end{smallmatrix}$ *is a fibration. It is said to be obtained from* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *by* ***change of base*** *along* f.

We will only use this construction in the case where $\mathbb{B}'$ is a full subcategory of $\mathbb{B}$. In this case the new fibration $\begin{smallmatrix}\mathbb{E}'\\ \downarrow p'\\ \mathbb{B}'\end{smallmatrix}$ is easy to describe: if $I \in \mathbb{B}'$, the fibre $\mathbb{E}'_I$ is just $\mathbb{E}_I$.

Example 2.2.13 Let $\mathcal{F} \hookrightarrow \mathcal{S}ets$ be the full subcategory of finite sets. The fibration $\begin{smallmatrix}\mathbb{E}\\ \downarrow\\ \mathcal{F}\end{smallmatrix}$ obtained from $\begin{smallmatrix}Fam(\mathcal{S}ets)\\ \downarrow p\\ \mathcal{S}ets\end{smallmatrix}$ by change of base along the inclusion functor $i\colon \mathcal{F} \hookrightarrow \mathcal{S}ets$ is equivalent to the fibration of *finite* families of sets: $\mathbb{E}$ is the full subcategory of Fam($\mathcal{S}ets$) whose objects are the families $\{X_i\}_{i\in I}$ where I is finite. ■

Before going on, we should look at the connection between fibrations and indexed categories. Let $\mathbb{B}$ be some arbitrary base category.

Definition 2.2.14 *A* ***$\mathbb{B}$-indexed category*** *is a pseudofunctor* $F\colon \mathbb{B}^{op} \to \mathcal{C}at$*; that is,* F *takes each* $I \in \mathbb{B}$ *to a category* $F(I)$*, and each map* $\alpha\colon K \to I$ *in* $\mathbb{B}$ *to a functor* $\alpha^*\colon F(I) \to F(K)$*, called a* ***reindexing functor*** *or* ***substitution functor****, such that*

- $1_I^* \cong 1_{F(I)}$*;*
- *if* $K \overset{\beta}{\to} J \overset{\alpha}{\to} I$*, then* $(\alpha\beta)^* \cong \beta^*\alpha^*$*; and*
- *these isomorphisms satisfy several coherence conditions, not specified here.*

The indexed category is called ***split*** *if* F *is a functor; i.e. if the isomorphisms are identities. (In this case, the coherence conditions become vacuous.)*

Intuitively, $F(I)$ is a category of I-indexed families, and the functors α^* reindex families along α. Note that split indexed categories rarely occur naturally, except for those which are constructed syntactically, or those arising as 'externalisations' of internal categories (see 3.8.25). So in general, we do have to worry about the coherence conditions. This is one reason why it's more convenient to work with fibrations, for which there are no coherence conditions.

Suppose $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ is a fibration *with a cleavage:* then it gives rise to an indexed category $F\colon \mathbb{B} \to \mathcal{C}at$ in the following way. We let

$$F(I) = \mathbb{E}_I = p^{-1}(I)$$

We need to show that each $\alpha\colon J \to I$ induces a functor $\alpha^*\colon \mathbb{E}_I \to \mathbb{E}_J$. Suppose $X \in \mathbb{E}_I$; then the cleavage gives us a chosen cartesian lifting

$$\begin{tikzcd}
\alpha^* X \arrow[r, dashed, "g_X"] \arrow[d, phantom, ""] & X \\
K \arrow[r, "\alpha"'] & I
\end{tikzcd}$$

If $f\colon Y \to X$ in $\mathbb{E}_I$, then by the universal property of the cartesian map g_X, there is a unique factorization

$$\begin{tikzcd}
\alpha^* Y \arrow[r, "g_Y"] \arrow[dr, dashed, "\alpha^* f"'] & Y \arrow[dr, "f"] & \\
& \alpha^* X \arrow[r, "g_X"'] & X \\
J \arrow[r, "\alpha"] \arrow[dr, "1_J"'] & I \arrow[dr, "1_I"] & \\
& J \arrow[r, "\alpha"'] & I
\end{tikzcd}$$

By the way, if $\mathbb{B}$ has a terminal object, so that given $I \in \mathbb{B}$ there is a unique map $!\colon I \to 1$, then we often write I^* for $!^*$. For example, in the fibration $\begin{smallmatrix}\mathbb{B}^{\to}\\ \downarrow cod\\ \mathbb{B}\end{smallmatrix}$,

$$I^*\begin{pmatrix} J \\ \downarrow \\ 1 \end{pmatrix} = \begin{pmatrix} I \times J \\ \downarrow \pi_I \\ I \end{pmatrix}$$

Exercise 2.2.15 Define what it means for two indexed categories to be equivalent. Show that different cleavages for a fibration $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ give rise to equivalent indexed categories. Must they give rise to *isomorphic* indexed categories? ■

Definition 2.2.16 *A cloven fibration is* ***split*** *(or, more precisely, the cleavage is a* ***splitting****) if the above process gives rise to a split indexed category.*

Every fibration over a category $\mathbb{B}$ is equivalent to a split fibration (there is an elegant proof due to John Power). However, it's not clear how to extend this result to more complicated structures on fibrations. This is the *coherence* problem, and as Curien observed, it comes up when we try to use fibrations to interpret type theories. There ought to be a nice 2-categorical approach to coherence problems for fibrations-with-structure, but it seems hard to generalise Power's methods.

The above construction has a converse: each indexed category $F\colon \mathbb{B} \to \mathit{Cat}$ gives rise to a cloven fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ via the *Grothendieck construction.* $\mathbb{E}$ is the following category:

objects: pairs $\langle I, X\rangle$ where $I \in \mathbb{B}$, $X \in F(I)$

arrows: a map $\langle I, X\rangle \to \langle J, Y\rangle$ is a pair $\langle \alpha, f\rangle$ where $\alpha\colon I \to J$ and $f\colon X \to \alpha^* Y$ in $F(I)$

In fact, the 2-categories of cloven fibrations over $\mathbb{B}$ and of $\mathbb{B}$-indexed categories are equivalent—in a certain 3-categorical sense. It's just a matter of taste which one we decide to use.

2.3 Completeness

It is easy to say what it means for a cloven fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ to have *finite limits*: each $\mathbb{E}_I$ should have finite limits, and each reindexing functor α^* should preserve them. However, as we noted in the introduction, these 'external' limits are not the ones we need to model type theories; rather, we have to say what it means for a fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ to have (say) I-indexed products, where $I \in \mathbb{B}$. A few preliminary examples will help motivate what follows, I hope.

Example 2.3.1 Let's see how to use fibrations (in a rather trivial way) to model the simply typed λ-calculus with type variables. In this case the types are

$$\tau \quad ::= \quad X \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2$$

How would we interpret this in a cartesian closed category $\mathbb{C}$? Type expressions should denote *families* of types; for example we should have

$$[\![X \to X]\!] = \{T^T \mid T \in \mathbb{C}_0\}$$

Similarly, terms should denote families of arrows. For example,

$$\begin{array}{rcl}[\![\lambda x : X.\, x]\!] & = & \{1 \xrightarrow{\ulcorner 1_X \urcorner} T^T \mid T \in \mathbb{C}_0\}\\ [\![vu\,[u : X \to Y, v : Y \to Z]]\!] & = & \{T^S \times U^T \xrightarrow{compose} U^S \mid S, T, U \in \mathbb{C}_0\}\end{array}$$

The fibration we need is very simple. Let Δ be the free category with finite products on a single object I: its objects are $1, I, I^2, I^3, \ldots$ (In fact, Δ is just the opposite of the category $\mathcal{F}$ of finite sets.) There is a finite product preserving functor $\Delta \to \mathcal{S}ets$ taking $I \mapsto \mathbb{C}_0$, so we can define a fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \Delta\end{array}$ by base change of $\begin{array}{c}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{array}$ along $\Delta \to \mathcal{S}ets$.

To spell this out a bit more, the objects of the fibre $\mathbb{E}_{I^n}$ are 'n-ary type constructors':

$$\mathbb{E}_{I^n} = \{U_{T_1,\ldots,T_n}\}_{T_1,\ldots,T_n \in \mathbb{C}_0} \qquad \text{where} \qquad U_{T_1,\ldots,T_n} \in \mathbb{C}_0$$

(arbitrary ones, mostly not definable in the calculus) and the arrows are families of maps

$$\{f_{T_1,\ldots,T_n}: U_{T_1,\ldots,T_n} \to V_{T_1,\ldots,T_n}\}_{T_1,\ldots,T_n}$$

Each fibre is a cartesian closed category, because products and exponentials can just be defined pointwise. Similarly, each reindexing functor α^* preserves the cartesian closed structure; so we have a fibred CCC. Note that this fibration is really much bigger than we need. ■

We could use a larger base category—for example, if $\mathbb{C}$ were small we could use the *full* subcategory of $\mathcal{S}ets$ on the finite powers of $\mathbb{C}_0$—but there is really no need to. Anyway, the question is: what extra structure, or extra properties, do we need in order to model System F? That is, how can we model types like

$$\forall X. (X \to X) \to (X \to X) \quad ?$$

The introduction and elimination rules for this polymorphic type suggests that we should interpret it as a product, indexed by $\mathbb{C}_0$. How can we make sense of this idea?

Example 2.3.2 Fibrations can also be used to model logic. Let $\mathcal{L}$ be the propositional language generated by some relation symbols $R(x), S(x,y)\ldots$ We will construct a fibration that models classical propositional logic over $\mathcal{L}$. First some notation: if $V = \{x, y, \ldots\}$ is a finite set of variables, let

$$\mathcal{L}(V) = \{\phi \in \mathcal{L} \mid FV(\phi) \subseteq V\}$$

Let the objects of $\mathbb{B}$ be finite sets of variables, and let a map $V' \to V$ in $\mathbb{B}$ be a function $\sigma: V \to V'$. Such a function σ defines a substitution map

$$\sigma^*: \mathcal{L}(V) \to \mathcal{L}(V')$$

For example, if $V = \{x, y\}$ and $V' = \{u, v, w\}$ then $\sigma^*(\phi(x,y)) = \phi(\sigma(x), \sigma(y))$. Note that $\mathbb{B}$ is actually equivalent to Δ.

Let $\mathbb{E}$ be the following category:

objects: pairs $\langle \phi, V\rangle$ with $\phi \in \mathcal{L}(V)$

arrows: $\langle \phi, V\rangle \to \langle \phi', V'\rangle$ is given by a map $V \to V'$
(a function $\sigma: V' \to V$) such that $\phi \models_V \sigma^*(\phi')$

Here $\models$ denotes classical entailment.

So $\mathbb{E}_V$ is just the preorder $\mathcal{L}(V)$, ordered by classical entailment; it is a pre-Boolean algebra. The substitution maps $\sigma^*: \mathbb{E}_V \to \mathbb{E}_{V'}$ preserve entailment and commute with all the logical operations, up to logical equivalence. The cartesian maps in $\mathbb{E}$ are (up to logical equivalence) arrows of the form $\langle \sigma^*(\phi'), V\rangle \to \langle \phi', V'\rangle$; that is, they are just the substitutions.

The logical operations $\phi \wedge \psi$, $\phi \vee \psi$ correspond to binary products and sums in each fibre. Suppose we wanted to allow first-order formulas like $\exists x.\phi$; do they have some kind of categorical property as well? Note that, since quantification binds variables, such a property must relate two *different* fibres.

Let $\phi \in \mathcal{L}(V), \psi \in \mathcal{L}(V \cup \{x\})$. Note that the inclusion $\sigma: V \hookrightarrow V \cup \{x\}$ defines a trivial substitution map $\sigma^*: \mathcal{L}(V) \to \mathcal{L}(V \cup \{x\})$; we will simply write ϕ for $\sigma^*(\phi)$, as it is the same formula. Now, it's easy to see that

$$\begin{array}{rcl} \phi \models_V \forall x.\, \psi & \text{iff} & \phi \models_{V \cup \{x\}} \psi \\ \exists x.\, \psi \models_V \phi & \text{iff} & \psi \models_{V \cup \{x\}} \phi \end{array}$$

This suggests that $\exists x \dashv \sigma^* \dashv \forall x$. Also, note that if we identify $\mathbb{B}$ with Δ, σ corresponds to a product projection $I^n \to I^{n-1}$. ■

Going back to our original question: what does it mean for a fibration $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ to have I-indexed products, where $I \in \mathbb{B}$? We suppose that $\mathbb{B}$ has finite products. Then given an object $X \in \mathbb{E}_I$, which we informally think of as a family $(X_i)_{i \in I}$, we'd like an object $\Pi_I X \in \mathbb{E}_1$ which we can think of as $\Pi_{i \in I} X_i$.

Let's see what this means for $\mathbb{E} = \mathrm{Fam}(\mathbb{C})$, where $\mathbb{C}$ is a category with ordinary (small) products. Suppose we have a family $\{X_i\}_{i \in I} \in \mathrm{Fam}(\mathbb{C})_I$. Then if $Y \in \mathbb{C}$, maps $f: Y \to \Pi_i X_i$ correspond to families of maps

$$\{f_i: Y \to X_i\}_{i \in I}$$

(this is just the universal property of the product). That is, they correspond to maps $I^*Y \to \{X_i\}_{i \in I}$ in $\mathbb{E}_I$. In other words, $\Pi_i(-): \mathbb{E}_I \to \mathbb{E}_1 = \mathbb{C}$ is right adjoint to $I^*: \mathbb{E}_1 \to \mathbb{E}_I$.

In fact, for any set J the functor $\Pi_i: \mathbb{E}_{I \times J} \to \mathbb{E}_J$ taking

$$\{X_{i,j}\}_{i \in I, j \in J} \mapsto \{\prod_i X_{i,j}\}_{j \in J}$$

is right adjoint to π^*, where $\pi: I \times J \to J$.

This suggests that we make the following definition. Let $\mathbb{B}$ be a category with finite products, let $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ be a cloven fibration over $\mathbb{B}$ and let $I \in \mathbb{B}$.

Definition 2.3.3 $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ ***has I-indexed products*** *if*

1. *For any $J \in \mathbb{B}$, if $\pi_{I,J}: I \times J \to J$ is the product projection, then $\pi^*_{I,J}$ has a right adjoint $\Pi_{I,J}: \mathbb{E}_{I \times J} \to \mathbb{E}_I$; and*

2. *The* ***Beck-Chevalley condition*** *holds: given any map $\alpha: K \to J$, giving us a pullback*

$$\begin{tikzcd} I \times K \arrow[r, "1_I \times \alpha"] \arrow[d, "\pi_{I,K}"'] \arrow[dr, phantom, "\lrcorner", very near start] & I \times J \arrow[d, "\pi_{I,J}"] \\ K \arrow[r, "\alpha"'] & J \end{tikzcd}$$

the canonical natural transformation

$$\alpha^* \Pi_{I,J} \Rightarrow \Pi_{I,K}(1_I \times \alpha)^*: \mathbb{E}_{I \times J} \to \mathbb{E}_K$$

is a natural isomorphism.

Although the definition refers to reindexing functors, it does not depend on a choice of cleavage; we'll see this in a moment.

First we should see where the canonical map in the Beck-Chevalley condition comes from. Because of the adjunctions $\pi^*_{I,J} \dashv \Pi_{I,J}$, given $X \in \mathbb{E}_{I\times J}$, there is a sequence of natural maps between homsets

$$\frac{\dfrac{\dfrac{\dfrac{\Pi_{I,J}X \longrightarrow \Pi_{I,J}X}{\pi^*_{I,J}\Pi_{I,J}X \longrightarrow X}}{(1_I\times\alpha)^*\pi^*_{I,J}\Pi_{I,J}X \longrightarrow (1_I\times\alpha)^*X}}{\pi^*_{I,K}\alpha^*\Pi_{I,J}X \longrightarrow (1_I\times\alpha)^*X}}{\alpha^*\Pi_{I,J}X \longrightarrow \Pi_{I,K}(1_I\times\alpha)^*X}$$

(we have used the fact that $\pi_{I,J}(1_I \times \alpha) = \alpha\pi_{I,K}$). So we can begin with the identity map at the top and work our way down to the bottom.

The Beck-Chevalley condition looks a little mysterious at first. There are at least two ways of understanding it. The first is to see what it means in the case $\mathbb{E} = \mathrm{Fam}(\mathbb{C})$. Suppose we have a family $\{X_{i,j}\}_{i\in I, j\in J} \in \mathrm{Fam}(\mathbb{C})_{I\times J}$; let's have a look at what the two functors do:

$$\begin{array}{rcccc} \alpha^*\Pi_{I,J}\colon & \{X_{i,j}\}_{i\in I,j\in J} & \mapsto & \{\prod_i X_{i,j}\}_{j\in J} & \mapsto & \{\prod_i X_{i,\alpha(k)}\}_{k\in K} \\ \Pi_{I,K}(1_I\times\alpha)^*\colon & \{X_{i,j}\}_{i\in I,j\in J} & \mapsto & \{X_{i,\alpha(k)}\}_{i\in I,k\in K} & \mapsto & \{\prod_i X_{i,\alpha(k)}\}_{k\in K} \end{array}$$

Clearly we should get the same answer. The slogan is, *quantification commutes with substitution.*

The second way of understanding the Beck condition is more categorical; it is just what's required for the adjunctions $\pi_{I,J} \dashv \Pi_{I,J}$ to fit together to form a fibred or 'global' adjunction, and is hence a natural coherence condition. The explanation goes as follows.

We define a new fibration $\begin{array}{c}\mathbb{E}\bullet I\\ \downarrow q\\ \mathbb{B}\end{array}$ as follows: $\mathbb{E}\bullet I$ is the category with

objects: X, where $X \in \mathbb{E}_{I\times J}$ for some $J \in \mathbb{B}$

arrows: same as in $\mathbb{E}$

(so that $(\mathbb{E}\bullet I)_J = \mathbb{E}_{I\times J}$); and we put $q(X) = J$. There is a cartesian functor

$$\begin{tikzcd} \mathbb{E} \arrow[rr, "\delta"] \arrow[dr, "p"'] & & \mathbb{E}\bullet I \arrow[dl, "q"] \\ & \mathbb{B} & \end{tikzcd}$$

Namely, if $Y \in \mathbb{E}_J$ and $\pi_{I,J}\colon I\times J \to J$ is the projection, then $\delta(Y) = \pi^*_{I,J}Y$. We now have the following result, for which the Beck-Chevalley condition is needed.

Proposition 2.3.4 *δ has a fibred right adjoint if and only if $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ has I-indexed products.*

Interestingly, although our original definition only applied to cloven fibrations, this new characterisation makes sense even if we haven't picked a cleavage for $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$. So in a way it is a more natural definition to make, though it's less easily understood.

Let's go back to the fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \Delta\end{array}$ from example 2.3.1, that was used to model the simply typed λ-calculus. It will be a model of System F exactly when it has I-indexed products.

To see why this is so, suppose $\tau(X,Y)$ and $\sigma(Y)$ are types. Then $[\![\tau(X,Y)]\!]$ is some family

$$\{T_{U,V} \mid U, V \in \mathbb{C}_0\}$$

The projection $\pi_{I,I}: I^2 \to I$ gives us a functor $\Pi_{I,I}: \mathbb{E}_{I^2} \to \mathbb{E}_I$, and this must take

$$\{T_{U,V}\}_{U,V\in\mathbb{C}_0} \mapsto \{\prod_{U\in\mathbb{C}_0} T_{U,V}\}_{V\in\mathbb{C}_0}$$

By the adjunction, there is a natural bijection

$$\frac{\pi_{I,I}^*[\![\sigma(Y)]\!] \longrightarrow [\![\tau(X,Y)]\!]}{[\![\sigma(Y)]\!] \longrightarrow \Pi_{I,I}[\![\tau(X,Y)]\!]}$$

so if we take $[\![\Pi X.\, \tau(X,Y)]\!] = \Pi_{I,I}[\![\tau(X,Y)]\!]$, we can interpret the introduction and elimination rules

$$t : \tau(X,Y)\,[x : \sigma(Y)] \quad \longleftrightarrow \quad \Lambda X.\, t : \Pi X.\, \tau(X,Y)\,[x : \sigma(Y)]$$

Unfortunately, I don't think there are any nontrivial examples of categories $\mathbb{C}$ closed under $\mathbb{C}_0$-indexed products. In effect, this means that in order to model polymorphic types, we have to look at fibrations which do *not* come from those of the form $\begin{array}{c}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{array}$. Vaguely, products will be 'too big to exist' unless maps between families are defined in a more 'uniform' way.

In order to model dependent types, we need more general kinds of completeness. A precise statement of the structure one needs is messy, but the following general definition covers most of the mathematical examples.

Definition 2.3.5 *Let $\mathbb{B}$ be a category with finite limits. A cloven fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ has $\mathbb{B}$-**indexed products** if*

1. *Given any map $\phi: J \to I$ in $\mathbb{B}$, ϕ^* has a right adjoint $\Pi_\phi: \mathbb{E}_J \to \mathbb{E}_I$; and*
2. *The **Beck-Chevalley condition** holds: given any pullback square*

$$\begin{tikzcd} L \arrow[r, "\beta"] \arrow[d, "\psi"'] \arrow[dr, phantom, "\lrcorner", very near start] & J \arrow[d, "\phi"] \\ K \arrow[r, "\alpha"'] & I \end{tikzcd}$$

in $\mathbb{B}$, the canonical natural transformation

$$\alpha^* \Pi_\phi \Rightarrow \Pi_\psi \beta^*: \mathbb{E}_J \to \mathbb{E}_K$$

is an isomorphism.

$\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ *is* ***complete*** *if it has* $\mathbb{B}$*-indexed products, each fibre* $\mathbb{E}_I$ *has finite limits and reindexing functors preserve finite limits.*

In this definition, we are implicitly regarding each map $\left(\begin{array}{c}J\\ \downarrow\phi\\ I\end{array}\right)$ as presenting an I-indexed family of objects of $\mathbb{B}$, and computing products indexed by each of these objects. That is, we're implicitly referring to the fibration $\begin{array}{c}\mathbb{B}^{\rightarrow}\\ \downarrow cod\\ \mathbb{B}\end{array}$.

Example 2.3.6 Consider the fibration $\begin{array}{c}Fam(\mathcal{S}ets)\\ \downarrow p\\ \mathcal{S}ets\end{array}$. A function $\phi: J \to I$ defines a family $\{\phi^{-1}\{i\}\}_{i\in I}$, and

$$\Pi_\phi: \{X_j\}_{j\in J} \mapsto \{\prod_{\phi(j)=i} X_j\}_{i\in I}$$

That is, we take products indexed by each of the sets $\phi^{-1}\{i\}$. So the fibration $\begin{array}{c}Fam(\mathcal{S}sets)\\ \downarrow p\\ \mathcal{S}ets\end{array}$ is complete. More generally, $\begin{array}{c}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{array}$ is a complete fibration iff $\mathbb{C}$ is complete (i.e. has all small limits). ■

Exercise 2.3.7 Give a 'cleavage-free' definition of completeness, involving the existence of fibred right adjoints to certain cartesian functors; it should be analogous to our alternative characterisation of closure under I-indexed products. (Clue: note that we could have defined the fibration $\begin{array}{c}\mathbb{E}\bullet I\\ \downarrow q\\ \mathbb{B}\end{array}$ by change of base—generalise.) ■

We close this section by mentioning indexed coproducts, which can be used to interpret existential types ('weak sums') in type theory. They are defined in exactly the same way as indexed products: this time, each reindexing functor ϕ^* is required to have a *left* adjoint Σ_ϕ that satisfies a Beck-Chevalley condition[3].

Exercise 2.3.8 Let $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ be a cloven fibration such that for all $\alpha: K \to I$ in $\mathbb{B}$, α^* has both a left adjoint Σ_α and a right adjoint Π_α. The functors Σ_α satisfy the Beck condition if and only if the functors Π_α satisfy the Beck condition. ■

Note that if $\mathbb{B}$ is a category with pullbacks, the fibration $\begin{array}{c}\mathbb{B}^{\rightarrow}\\ \downarrow cod\\ \mathbb{B}\end{array}$ always has $\mathbb{B}$-indexed coproducts. Namely, if $\left(\begin{array}{c}J\\ \downarrow\phi\\ K\end{array}\right) \in \mathbb{B}/K = (\mathbb{B}^{\rightarrow})_K$ and $\alpha: K \to I$, then

$$\Sigma_\alpha\left(\begin{array}{c}J\\ \downarrow\phi\\ K\end{array}\right) = \left(\begin{array}{c}J\\ \downarrow\alpha\phi\\ I\end{array}\right)$$

That is, 'coproducts are computed by composition'. In fact, it turns out that in this situation one can actually model strong sums in dependent type theory. But that's another story.

[3]Note that the canonical map goes in the opposite direction.

Exercise 2.3.9 Let $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ be a cloven fibration with $\mathbb{B}$-indexed coproducts. Given $\alpha: K \to I$, $X \in \mathbb{E}_I$, and $Z \in \mathbb{E}_K$, show how to define a natural map

$$\Sigma_\alpha(Z \times \alpha^* X) \to \Sigma_\alpha Z \times X$$

If this map is an isomorphism, we say that **Frobenius reciprocity** holds. Suppose that each of the fibres $\mathbb{E}_I$ is cartesian closed. Show that Frobenius reciprocity holds if and only if each reindexing functor α^* preserves exponentials, i.e. if $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ is a fibred CCC. ■

The logical intuition behind Frobenius reciprocity is that

$$\exists x.\,(\psi(x,y) \wedge \phi(y)) \dashv\vdash (\exists x.\,\psi(x,y)) \wedge \phi(y)$$

In this form, it is discussed in [9, remark 1.5].

2.4 Smallness

Let's recall the view that a fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ describes a category 'over' or 'relative to' $\mathbb{B}$. We've just seen what it means for $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ to have products indexed by objects of $\mathbb{B}$. What does it mean for $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ to have a small 'set' of objects, or to be locally small, relative to $\mathbb{B}$? These are important questions: smallness (of the 'set' of objects) together with completeness lets us model polymorphism, while local smallness lets us make sense of dependent types.

First we'll say what it means for a fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ to 'have a small set of objects' relative to $\mathbb{B}$. The definition is surprisingly simple.

Definition 2.4.1 *$G \in \mathbb{E}_P$ is a **generic object** for $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ if for all $X \in \mathbb{E}_I$ there is a cartesian map $X \to G$; equivalently, if there is a map $\phi: I \to P \in \mathbb{B}$ such that $X \cong \phi^* G$. (We implicitly assume that we are given a* choice *of cartesian map $X \to G$.)*

We regard P as the 'set' of objects of the 'category over $\mathbb{B}$' which the fibration $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ describes; G is the 'family of all the objects in the category'. The idea, then, is that to give some $X \in \mathbb{E}_I$—thought of, informally, as an I-indexed family (X_i)—it suffices to give a map $\chi: I \to P$ taking $i \mapsto X_i$. X is then (isomorphic to) $\chi^* G$; that is, χ picks out the objects we want.

Example 2.4.2 Let $\mathbb{C}$ be a category with a small set of objects; we will show that $\begin{array}{c}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{array}$ has a generic object. Let $P = \mathbb{C}_0$ and let $G \in \mathbb{E}_P$ be $\{X\}_{X \in \mathbb{C}_0}$. Any family $\{X_i\}_{i \in I} \in \mathrm{Fam}(\mathbb{C})$ defines a function

$$\phi: I \to \mathbb{C}_0$$

namely $\phi(i) = X_i$. This function ϕ, together with the identity maps $1_{X_i}: X_i \to X_i$, defines a cartesian map $\{X_i\}_{i \in I} \to G = \{X\}_{X \in \mathbb{C}_0}$. ■

Example 2.4.3 After the discussion in the last section, it should be plausible that models of System F correspond to cloven fibrations $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ such that:

1. $\mathbb{B}$ has finite products;
2. $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ is a fibred CCC;
3. There is a generic object $G \in \mathbb{E}_P$; and
4. $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ has P-indexed products.

In terms of example 2.3.1, we would have $P = I$.

The interpretation of a single type variable X is just $G \in \mathbb{E}_P$; so, for example, $[\![X \to X]\!] = G^G \in \mathbb{E}_P$. More generally, the interpretation of a type $\tau(X_1, \ldots, X_n)$ with n free type variables is some object

$$[\![\tau(X_1, \ldots, X_n)]\!] \in \mathbb{E}_{P^n}$$

Polymorphic types are interpreted as P-indexed products; e.g.

$$[\![\Pi X.\, X \to X]\!] = \Pi_{P,1} G^G$$

The adjunctions $\pi^*_{P,J} \dashv \Pi_{P,J}$ allow us to model the introduction and elimination rules for polymorphic types. For example, consider the following deduction:

$$\cfrac{\cfrac{x : X \vdash x : X}{\vdash \lambda x : X.\, x : X \to X}}{\vdash \Lambda X.\, \lambda x : X.\, x : \Pi X.\, X \to X}$$

This gives us a recipe for constructing an arrow in the following way:

$$\cfrac{\cfrac{1_G \colon G \to G}{\ulcorner 1_G \urcorner \colon \pi^*_{P,1} 1 \cong 1 \to G^G}}{\overline{\ulcorner 1_G \urcorner} \colon 1 \to \Pi_{P,1} G^G}$$

■

There is no reason why a generic object should be unique up to isomorphism, as there is no uniqueness clause in the definition. To get uniqueness up to isomorphism, then, we simply add a uniqueness clause. Unfortunately this now excludes many interesting examples.

Definition 2.4.4 *$G \in \mathbb{E}_P$ is a **skeletal generic object** for $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ if for all $X \in \mathbb{E}$ there is a unique cartesian map $X \to G$.*

Exercise 2.4.5 Explain why, in example 2.4.2, G need not be a skeletal generic object. Observe that it is skeletal precisely when $\mathbb{C}$ is *skeletal*, i.e. when the only isomorphisms in $\mathbb{C}$ are the identity maps. ■

Example 2.4.6 Let Mono($\mathcal{S}ets$) be the full subcategory of $\mathcal{S}ets^{\to}$ whose objects are the monomorphisms. Then $\begin{array}{c} Mono(\mathcal{S}ets) \\ \downarrow cod \\ \mathcal{S}ets \end{array}$ is a full subfibration of $\begin{array}{c} \mathcal{S}ets^{\to} \\ \downarrow cod \\ \mathcal{S}ets \end{array}$. Mono$(\mathcal{S}ets)_X$ is (equivalent to) the collection of subsets of X, ordered by inclusion.

Let $1 = \{*\}, 2 = \{tt, ff\}$ and $\top\colon 1 \rightarrowtail 2$ be the map taking $* \mapsto tt$. Then the object $\left(\begin{array}{c} 1 \\ \downarrow\top \\ 2 \end{array} \right) \in$ Mono$(\mathcal{S}ets)_2$ is a skeletal generic object in the fibration $\begin{array}{c} Mono(\mathcal{S}ets) \\ \downarrow cod \\ \mathcal{S}ets \end{array}$. For given any set X, a monomorphism into X may be assumed to be a subset inclusion $X' \hookrightarrow X$; and there is a pullback square

$$\begin{tikzcd} X' \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, "\top"] \\ X \arrow[r, "\chi_{X'}"'] & 2 \end{tikzcd}$$

where $\chi_{X'}$ is the characteristic function of X'. That is, subsets of X correspond to predicates $X \to 2$. ■

Example 2.4.7 Let B be a Boolean algebra, thought of as a category (via its poset structure). Consider the fibration $\begin{array}{c} Fam(B) \\ \downarrow p \\ \mathcal{S}ets \end{array}$; the fibre over a set X is the poset B^X, the Boolean algebra of 'B-valued predicates' on X. Show that this fibration has a skeletal generic object. Show that it's complete iff B is a complete Boolean algebra. This line of thought eventually leads to tripos theory: see [9, 1.6]. ■

Exercise 2.4.8 Let $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ be a fibration equipped with a choice of cleavage. Say that $G \in \mathbb{E}_P$ is a **strict generic object** if for all $X \in \mathbb{E}_I$ there is a map $\phi\colon I \to P$ such that $X = \phi^* I$. What is a 'skeletal' strict generic object? Show that $\begin{array}{c} Fam(\mathbb{C}) \\ \downarrow p \\ \mathcal{S}ets \end{array}$ always has one. ■

Locally small fibrations

Defining local smallness is a bit more difficult, but the idea is similar: a fibration $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ will be locally small if, given any $X, Y \in \mathbb{E}_I$ there is a 'generic' family of maps $X \to Y$ (over an object $[X, Y] \in \mathbb{B}$, the 'set of maps from X to Y') such that any other map $X \to Y$ in any fibre is obtained by reindexing the generic family. Let's look at a simple example before giving the formal definition.

Consider the fibration $\begin{array}{c} Fam(\mathcal{S}ets) \\ \downarrow p \\ \mathcal{S}ets \end{array}$. Let $\{X\}, \{Y\} \in$ Fam$(\mathcal{S}ets)_1$; i.e. we are just given two single sets X, Y. Then the function space $Y^X \in \mathcal{S}ets$ parametrises functions $X \to Y$ (i.e. maps $\{X\} \to \{Y\}$ in Fam$(\mathcal{S}ets)_1$).

In fact, the family of functions $\{f\colon X \to Y\}$, which is indexed by the set Y^X, gives us a map

$$u\colon (Y^X)^* X \to (Y^X)^* Y$$

in $\mathrm{Fam}(\mathcal{S}ets)_{Y^X}$: it's 'the family of *all* maps $\{X\} \to \{Y\}$'. Any particular map $f\colon \{X\} \to \{Y\}$ in $\mathrm{Fam}(\mathcal{S}ets)_1 \simeq \mathcal{S}ets$ (i.e. any function $f\colon X \to Y$) gives rise to a function $\ulcorner f \urcorner\colon 1 \to Y^X$ 'naming' f, and $f = \ulcorner f \urcorner^* u$.

More generally, suppose I is a set and we have a vertical map

$$f\colon I^*\{X\} \to I^*\{Y\}$$

given by a family of maps $\{f_i\colon X \to Y\}_{i \in I}$; then we get a function $\ulcorner f \urcorner\colon I \to Y^X$ taking $i \mapsto f_i$, and $f = \ulcorner f \urcorner^* u$ again.

Definition 2.4.9 *A cloven fibration* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *is* ***locally small*** *if for each* $I \in \mathbb{B}$ *and each* $X, Y \in \mathbb{E}_I$*, there is an object* $[X,Y] \in \mathbb{B}$*, a map* $\pi\colon [X,Y] \to I$ *and a map*

$$\begin{tikzcd} & \pi^* X \arrow[r, "u"] & \pi^* Y \arrow[d, dotted, "p"] \\ I & & {[X,Y]} \arrow[ll, "\pi"] \end{tikzcd}$$

which is a ***generic map*** $X \to Y$*: that is, given any* $\alpha\colon J \to I$*, and any map*

$$f\colon \alpha^* X \to \alpha^* Y \ \in \mathbb{E}_J$$

there is a unique map $\ulcorner f \urcorner\colon J \to [X,Y]$ *in* $\mathbb{B}$ *such that* $f = \ulcorner f \urcorner^* u$.

When $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix} = \begin{smallmatrix}Fam(\mathcal{S}ets)\\ \downarrow p\\ \mathcal{S}ets\end{smallmatrix}$, this is the situation just described.

More concisely, it can be shown that $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ is locally small exactly when, for all $I \in \mathbb{B}$, the functor $(\mathbb{B}/I)^{op} \to \mathcal{S}ets$ taking

$$\left(\begin{smallmatrix}J\\ \downarrow \alpha\\ I\end{smallmatrix}\right) \mapsto \mathbb{E}_J(\alpha^* X, \alpha^* Y)$$

is representable. (At least, this is equivalent to the original definition so long as $\mathbb{E}$ is locally small in the ordinary sense, so that this functor can actually be defined.) Note that this alternative definition does not refer to a cleavage for the fibration.

Example 2.4.10 If $\mathbb{C}$ is a locally small category, $\begin{smallmatrix}Fam(\mathbb{C})\\ \downarrow p\\ \mathcal{S}ets\end{smallmatrix}$ is a locally small fibration. For example, if $X, Y \in \mathbb{C}$, so that $\{X\}, \{Y\} \in \mathrm{Fam}(\mathbb{C})_1$, then

$$[\{X\}, \{Y\}] = \mathbb{C}(X, Y)$$

In general, suppose $\{X_i\}_{i \in I}, \{Y_i\}_{i \in I} \in \mathrm{Fam}(\mathbb{C})_I$; then we take the disjoint union of the homsets:

$$[\{X_i\}_{i \in I}, \{Y_i\}_{i \in I}] = \coprod_{i \in I} \mathbb{C}(X_i, Y_i) \xrightarrow{\pi} I$$

where π takes $f\colon X_i \to Y_i$ to i. In each case, u is given by the obvious family of maps. ■

So in this case, the clause in the definition of local smallness saying that $\ulcorner f \urcorner$ should be unique is quite natural, and doesn't exclude interesting examples (like the definition of skeletal generic object does). This reflects the fact that it is natural to speak of equality of arrows in a category, but not equality of objects.

Example 2.4.11 The first fibration of subsets of $\mathbb{N}$ defined in example 2.2.8 is locally small. However, the second one is not, because the maps are defined differently.

To see this, consider the singleton family $\{\mathbb{N}\} \in \mathbb{E}_1$, and let $f\colon 2^*\{\mathbb{N}\} \to 2^*\{\mathbb{N}\}$ in $\mathbb{E}_2$. If $\mathsf{in}_0, \mathsf{in}_1\colon 1 \to 1+1 = 2$ are the coproduct inclusions, then

$$\mathsf{in}_0^* f = \mathsf{in}_1^* f$$

This is because of the 'uniform' way maps in $\mathbb{E}$ are defined: they come from a single function $\mathbb{N} \to \mathbb{N}$, rather than being arbitrary families.

Now, suppose the fibration were locally small. Then we would have a set $[\{\mathbb{N}\},\{\mathbb{N}\}]$ and a generic map

$$u\colon [\{\mathbb{N}\},\{\mathbb{N}\}]^*\{\mathbb{N}\} \longrightarrow [\{\mathbb{N}\},\{\mathbb{N}\}]^*\{\mathbb{N}\}$$

Let $g, h\colon \mathbb{N} \to \mathbb{N}$ be two different functions, defining two different maps $\{\mathbb{N}\} \to \{\mathbb{N}\}$ in $\mathbb{E}_1$. These give rise to $\ulcorner g \urcorner, \ulcorner h \urcorner\colon 1 \to [\{\mathbb{N}\},\{\mathbb{N}\}]$. Let $\phi\colon 2 \to [\{\mathbb{N}\},\{\mathbb{N}\}]$ be the unique map such that

$$\begin{tikzcd}
1 \arrow[d, "\mathsf{in}_0"'] \arrow[dr, "\ulcorner g \urcorner"] & \\
2 \arrow[r, "\phi"] & {[\{\mathbb{N}\},\{\mathbb{N}\}]} \\
1 \arrow[u, "\mathsf{in}_1"] \arrow[ur, "\ulcorner h \urcorner"'] &
\end{tikzcd}$$

Then, up to isomorphism,

$$\mathsf{in}_0^* \phi^* u = \ulcorner g \urcorner^* u = g \neq h = \ulcorner h \urcorner^* u = \mathsf{in}_1^* \phi^* u$$

which contradicts the previous paragraph (put $f = \phi^* u$). ■

The same applies to PERs. To use them to model polymorphic type theories, we have to construct a fibration out of them. But modelling dependent types requires local smallness, as we'll see in a moment. So it's sometimes necessary to think of them as fibred over ω-sets instead of just $\mathcal{S}ets$: the fibration of PERs over $\mathcal{S}ets$ is not locally small. See 4.2.16.

There is a close relationship between local smallness and locally cartesian closed categories, which are the usual setting used to model Martin-Löf type theory. (We will see how this works in the next chapter.) This helps to explain the relevance of local smallness to modelling type dependency.

Definition 2.4.12 *A category $\mathbb{B}$ is* ***locally cartesian closed*** *if each slice $\mathbb{B}/I$ is a cartesian closed category.*

Let $\mathbb{B}$ be a locally cartesian closed category with a terminal object. Note that $\mathbb{B}$ must have finite limits, and in particular pullbacks; so $\begin{array}{c}\mathbb{B}^{\to}\\ \downarrow cod\\ \mathbb{B}\end{array}$ is a fibred CCC. Recall that this fibration has $\mathbb{B}$-indexed coproducts.

Exercise 2.4.13 For any $\alpha\colon K \to I$ in $\mathbb{B}$, $\alpha^*\colon \mathbb{B}/I \to \mathbb{B}/K$ preserves finite limits and exponentials. Also, $\begin{array}{c}\mathbb{B}^{\to}\\ \downarrow cod\\ \mathbb{B}\end{array}$ has $\mathbb{B}$-indexed products. (For the Beck condition, use 2.3.8.) ∎

Proposition 2.4.14 *Let $\mathbb{B}$ be a category with a terminal object and pullbacks. Then $\mathbb{B}$ is locally cartesian closed if and only if the fibration $\begin{array}{c}\mathbb{B}^{\to}\\ \downarrow cod\\ \mathbb{B}\end{array}$ is locally small.*

Proof. We'll sketch the 'only if' direction. Suppose $\mathbb{B}$ is locally cartesian closed; then given $X, Y \in \mathbb{B}$, we can define

$$[\left(\begin{array}{c}X\\ \downarrow\\ 1\end{array}\right), \left(\begin{array}{c}Y\\ \downarrow\\ 1\end{array}\right)] = Y^X$$

More generally, suppose we have $\left(\begin{array}{c}X\\ \downarrow\phi\\ I\end{array}\right), \left(\begin{array}{c}Y\\ \downarrow\psi\\ I\end{array}\right) \in \mathbb{B}/I$, which we think of as families $(X_i), (Y_i)$; then informally, we would expect

$$[\left(\begin{array}{c}X\\ \downarrow\phi\\ I\end{array}\right), \left(\begin{array}{c}Y\\ \downarrow\psi\\ I\end{array}\right)] = \coprod_i {Y_i}^{X_i}$$

Making this intuition precise is slightly messy.

Let $\left(\begin{array}{c}Z\\ \downarrow\zeta\\ I\end{array}\right) = \left(\begin{array}{c}Y\\ \downarrow\\ I\end{array}\right)^{\left(\begin{array}{c}X\\ \downarrow\\ I\end{array}\right)} \in \mathbb{B}/I$; intuitively, it is the family (Z_i) where $Z_i = {Y_i}^{X_i}$. Then we put

$$[\left(\begin{array}{c}X\\ \downarrow\\ I\end{array}\right), \left(\begin{array}{c}Y\\ \downarrow\\ I\end{array}\right)] = Z$$

We need to construct a generic map

$$u\colon \zeta^*\left(\begin{array}{c}X\\ \downarrow\\ I\end{array}\right) \to \zeta^*\left(\begin{array}{c}Y\\ \downarrow\\ I\end{array}\right) \in \mathbb{B}/Z$$

Now, the terminal object of $\mathbb{B}/Z$ is the identity map $\left(\begin{array}{c}Z\\ \downarrow 1_Z\\ Z\end{array}\right)$. So it suffices to construct a map

$$\left(\begin{array}{c}Z\\ \downarrow\\ Z\end{array}\right) \to \zeta^*\left(\begin{array}{c}Y\\ \downarrow\\ I\end{array}\right)^{\zeta^*\left(\begin{array}{c}X\\ \downarrow\\ I\end{array}\right)} \cong \zeta^*(\left(\begin{array}{c}Y\\ \downarrow\\ I\end{array}\right)^{\left(\begin{array}{c}X\\ \downarrow\\ I\end{array}\right)}) \cong \zeta^*\left(\begin{array}{c}Z\\ \downarrow\\ I\end{array}\right) \cong \left(\begin{array}{c}Z\times_I Z\\ \downarrow\pi_0\\ Z\end{array}\right)$$

where the final map appears in the pullback square

$$\begin{tikzcd}
Z \times_I Z \arrow[r, "\pi_1"] \arrow[d, "\pi_0"'] \arrow[dr, phantom, "\lrcorner", very near start] & Z \arrow[d, "\zeta"] \\
Z \arrow[r, "\zeta"'] & I
\end{tikzcd}$$

The desired map just comes from the diagonal map $\delta: Z \to Z \times_I Z$. Thought of as a map in $\mathbb{B}/Z$, this map intuitively represents the family of maps $\{z: 1 \to Z\}_{z \in Z}$. ∎

In toposes, we can phrase this argument in a much more attractive way: see 3.8.6.

The following definition is a specialisation of the one in Bart Jacobs' thesis. Suppose we have a fibred CCC $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ where $\mathbb{B}$ is locally cartesian closed.

Definition 2.4.15 *A* ***comprehension category structure*** *on* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *consists of a cartesian functor* $\mathcal{P}$: $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix} \longrightarrow \begin{smallmatrix}\mathbb{B}^{\to}\\ \downarrow cod\\ \mathbb{B}\end{smallmatrix}$. *(That is, a functor* $\mathcal{P}: \mathbb{E} \to \mathbb{B}^{\to}$ *over* $\mathbb{B}$ *that sends cartesian maps to pullback squares.)*

Proposition 2.4.16 $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ *is locally small if and only if there exists a comprehension category structure* $\mathcal{P}: \mathbb{E} \to \mathbb{B}^{\to}$ *on* $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$.

I won't try to describe what role comprehension categories play in the semantics of type theory. But let's at least see how local smallness suggests a way of modelling dependent types. Suppose we have a dependent type

$$\tau(x)\,[x : \sigma]$$

which we want to interpret in a fibration $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$. To do this, we need to be able to regard $[\![\sigma]\!]$ as an object of $\mathbb{B}$; then we can look for a suitable object $[\![\tau(x)]\!] \in \mathbb{E}_{[\![\sigma]\!]}$. Typically, however, $[\![\sigma]\!]$ is an object of $\mathbb{E}$, not $\mathbb{B}$; for example, if σ is a closed type then $[\![\sigma]\!] \in \mathbb{E}_1$.

However, if $\begin{smallmatrix}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{smallmatrix}$ is locally small (and $\mathbb{E}_1$ has a terminal object) then we can form the object $[1, [\![\sigma]\!]] \in \mathbb{B}$, and it makes sense to say that

$$[\![\tau(x)]\!] \in \mathbb{E}_{[1,[\![\sigma]\!]]}$$

Note than we also have the map $\pi: [1, [\![\sigma]\!]] \to 1$. So we can interpret dependent products as indexed products along this map:

$$[\![\Pi x : \sigma.\, \tau(x)]\!] = \Pi_\pi [\![\tau(x)]\!]$$

Exercise 2.4.17 What if σ is itself a dependent type? ∎

2.5 Definability

We haven't yet developed enough of the theory of fibrations to do category theory over a base category $\mathbb{B}$. In ordinary category theory, we often use an informal comprehension schema to define families of objects or arrows. The counterpart of this in the theory of fibrations is the notion of *definability*.

Let $\mathbb{C}$ be an ordinary category. Suppose $\mathbf{D} \subseteq \mathbb{C}_0$; then given any family $\{X_i\}_{i \in I}$ of objects of $\mathbb{C}$, it makes sense to talk about the set

$$\{i \in I \,|\, X_i \in \mathbf{D}\}$$

Similarly, suppose $X, Y \in \mathbb{C}$; then it makes sense to talk about the set

$$\{f: X \to Y \mid f \text{ is a mono}\}$$

What are the corresponding constructions for fibrations $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$? It turns out that they are not necessarily available, and that to insist that they are means imposing a rather strong condition on $\mathbb{B}$.

First some preliminaries. A class $\mathbf{D}$ of objects of $\mathbb{C}$ determines a class $\mathcal{D}$ of objects of $\mathrm{Fam}(\mathbb{C})$: we say that $\{X_i\}_{i\in I} \in \mathcal{D}$ when each $X_i \in \mathbf{D}$. As a class of objects of $\mathrm{Fam}(\mathbb{C})$, $\mathcal{D}$ has a special property: it is 'stable under reindexing', in the sense that if $\alpha: K \to I$ is any function, then the reindexed family $\{X_{\alpha(k)}\}_{k\in K}$ is also in $\mathcal{D}$. In fact, as long as $\mathbf{D}$ is closed under isomorphism, if $\{X_i\}_{i\in I} \in \mathcal{D}$ and we have any cartesian map

$$\{Y_j\}_{j\in J} \longrightarrow \{X_i\}_{i\in I}$$

then $\{Y_j\}_{j\in J} \in \mathcal{D}$.

Let $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ be a fibration and let $\mathcal{D}$ be a class of objects of $\mathbb{E}$.

Definition 2.5.1 *Definability for classes of objects:*

1. *$\mathcal{D}$ is **stable** if, whenever $X \in \mathcal{D}$ and $Y \to X$ is cartesian, $Y \in \mathcal{D}$.*

2. *If $\mathcal{D}$ is stable, $\mathcal{D}$ is **definable** if, for any $X \in \mathbb{E}$, there is an object $X_{|\mathcal{D}} \in \mathcal{D}$ and a cartesian map $i: X_{|\mathcal{D}} \to X$ with the following universal property: if $f: Y \to X$ is any cartesian map with $Y \in \mathcal{D}$, there is a unique cartesian map $g: Y \to X_{|\mathcal{D}}$ such that $f = ig$.*

If $I = pX$, we sometimes write $I_{X\in\mathcal{D}} = pX_{|\mathcal{D}}$.

The stable classes $\mathcal{D}$ of 'families' are intuitively the ones that come from classes $\mathbf{D}$ of objects, as above. If the fibration is cloven, stability just means that if $X \in \mathcal{D}$ then $\alpha^* X \in \mathcal{D}$. Note that if $\mathcal{D}$ is a stable class, it makes sense to talk about the **full subfibration** on the 'families' in $\mathcal{D}$.

As for definability: if we think of $X \in \mathbb{E}_I$ as being a family $(X_i)_{i\in I}$, then informally $X_{|\mathcal{D}}$ is the subfamily consisting of those X_i's which are elements of $\mathbf{D}$, and $I_{X\in\mathcal{D}}$ is the 'set' $\{i \in I \mid X_i \in \mathbf{D}\}$. The picture is:

$$\begin{tikzcd}
Y \arrow[dr, dashed, "\exists!\, g \text{ cartesian}"'] \arrow[drr, "f \text{ cartesian}"] & & \\
& X_{|\mathcal{D}} \arrow[r, "i"'] \arrow[dd, dotted, "p"] & X \\
\bullet \arrow[dr, dashed] \arrow[drr] & & \\
& I_{X\in\mathcal{D}} \arrow[r] & I
\end{tikzcd}$$

Note that the map $I_{X\in\mathcal{D}} \to I$ is a mono. Also note that we only defined definability for stable classes $\mathbb{D}$.

Example 2.5.2 Consider the fibration $\begin{array}{c} Fam(\mathcal{S}ets) \\ \downarrow p \\ \mathcal{S}ets \end{array}$. Define a class $\mathcal{D}$ of objects of Fam($\mathcal{S}ets$) by saying that a family $\{X_i\}_{i \in I} \in \mathcal{D}$ when each X_i is finite; this is the class of families arising from the class $\mathcal{F} \subset \mathcal{S}ets$ of finite sets.

$\mathcal{D}$ is definable. For given an arbitrary family $\{X_i\}_{i \in I}$, one can form the subfamily

$$\{X_i \mid X_i \text{ is finite}\}$$

indexed by the set $\{i \in I \mid X_i \text{ is finite}\}$. ■

I'm not sure what role definability of classes of objects might play in modelling type theories. However, it is a useful notion when we're working in richer frameworks. It's also of considerable theoretical importance, as Bénabou explains.

Exercise 2.5.3 If $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ is a fibration, formulate analogous notions of stability and definability for classes of arrows in $\mathbb{E}$. ■

There are other notions of definability. For example, let $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ be a cloven fibration.

Definition 2.5.4 *We say that* ***equality of arrows is definable*** *in* $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ *if for all* $I \in \mathbb{B}$*,* $X, Y \in \mathbb{E}_I$ *and* $f, g: X \to Y$*, there is an object* $I_{f=g} \in \mathbb{B}$ *and a map*

$$\iota: I_{f=g} \to I$$

such that $\iota^* f = \iota^* g = e$*, say; and given any map* $\alpha: K \to I$ *with* $\alpha^* f = \alpha^* g = h$*,* α *factors uniquely through* ι*.*

The picture is:

$$\begin{tikzcd}
\alpha^* X \arrow[r, "h"] & \alpha^* Y & & & \\
& & \iota^* X \arrow[r, "e"] & \iota^* Y \arrow[dd, dotted, "p"] & X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \\
& K \arrow[rrr, "\alpha"] \arrow[dr, dashed, "\exists!\, \phi"'] & & & \\
& & I_{f=g} \arrow[rr, "\iota"'] & & I &
\end{tikzcd}$$

so that $h = \phi^* e$.

Loosely, thinking of f and g as families of maps $(f_i: X_i \to Y_i)$, $(g_i: X_i \to Y_i)$, we can think of $I_{f=g}$ as being the 'set' $\{i \in I \mid f_i = g_i\} \subseteq I$.

Exercise 2.5.5 Find another definition of what it means for equality of arrows to be definable in a fibration, that doesn't refer to a cleavage. ■

Chapter 3

Toposes

3.1 Preliminaries

Before beginning our study of topos theory, we should briefly review constructive logic[1]. Just as Boolean algebras model classical propositional logic, there is a class of lattices—the *Heyting algebras*—which model constructive propositional logic. In a sense, topos theory generalises this, to give us an interpretation of higher-order constructive logic; in other words, 'constructive set theory'. The material in this section will not really be needed till we come to discuss the internal language of a topos, so you may want to put off reading it till then.

Propositional logic

Here is a proof system for constructive propositional logic. It follows the rules of natural deduction. However, instead of writing proofs as natural deduction trees, with hypotheses at the leaves, we use sequents to keep track of the hypotheses at each stage. This turns out to be convenient later on, but it does mean we have to add some structural rules to help with the bookkeeping.

Logical connectives

$$\phi \quad ::= \quad a\,(a \text{ atomic}) \,|\, \mathsf{true} \,|\, \mathsf{false} \,|\, \phi \wedge \psi \,|\, \phi \vee \psi \,|\, \phi \to \psi$$

"$\neg\phi$" is syntactic sugar for $\phi \to \mathsf{false}$.

We let $\mathcal{L}_A$ be the set of formulas generated by a set A of atomic propositions. For the moment, we don't care what the atomic propositions are. (In the internal language of a topos, they're all of the form $t = t'$.) $\vec{\phi}$ will denote a list $\phi_1, \ldots, \phi_n$ of propositions.

Start rules

$$\frac{}{\psi \vdash \psi} \qquad \frac{}{\vdash \mathsf{true}} \qquad \frac{}{\mathsf{false} \vdash \psi}$$

[1]Constructive logic is sometimes called *intuitionistic* logic. But it's handy to make the distinction between constructive mathematical reasoning, which is acceptable to classically minded mathematicians, and intuitionistic mathematics, which uses additional principles that are inconsistent with classical mathematics. Not all toposes are models for Brouwer's intuitionistic mathematics—some intuitionists would say that none of them are. So we will stick to the more neutral word 'constructive'.

Structural rules

$$\frac{\vec{\phi}, \phi_0, \phi_1, \vec{\phi'} \vdash \psi}{\vec{\phi}, \phi_1, \phi_0, \vec{\phi'} \vdash \psi} \qquad \frac{\vec{\phi} \vdash \psi}{\vec{\phi}, \phi' \vdash \psi} \qquad \frac{\vec{\phi}, \phi', \phi' \vdash \psi}{\vec{\phi}, \phi' \vdash \psi}$$

Logical rules

$$\frac{\vec{\phi} \vdash \psi \quad \vec{\phi} \vdash \psi'}{\vec{\phi} \vdash \psi \wedge \psi'} \qquad \frac{\vec{\phi} \vdash \psi \wedge \psi'}{\vec{\phi} \vdash \psi} \qquad \frac{\vec{\phi} \vdash \psi \wedge \psi'}{\vec{\phi} \vdash \psi'}$$

$$\frac{\vec{\phi} \vdash \psi}{\vec{\phi} \vdash \psi \vee \psi'} \qquad \frac{\vec{\phi} \vdash \psi'}{\vec{\phi} \vdash \psi \vee \psi'} \qquad \frac{\vec{\phi} \vdash \psi \vee \psi' \quad \vec{\phi}, \psi \vdash \zeta \quad \vec{\phi}, \psi' \vdash \zeta}{\vec{\phi} \vdash \zeta}$$

$$\frac{\vec{\phi}, \psi \vdash \psi'}{\vec{\phi} \vdash \psi \to \psi'} \qquad \frac{\vec{\phi} \vdash \psi \quad \vec{\phi} \vdash \psi \to \psi'}{\vec{\phi} \vdash \psi'}$$

This system is is meant to characterise the deductions

$$\phi_1, \ldots \phi_n \vdash \psi$$

that are 'constructively' valid. We can think of them, loosely, as the logical inferences which have some computational justification. One could make this more precise (for example, via realizability as in the next chapter, or within the 'propositions-as-types' paradigm); but perhaps a couple of examples will do for now.

Example 3.1.1 The constructive meaning of $\phi \vee \psi$ is: there is an *effective procedure* that tells us which of ϕ, ψ is true (if they're both true, it can give us either one). Classically,

$$\phi \vee \neg\phi$$

is a tautology. But constructively, $\vdash \phi \vee \neg\phi$ isn't provable. To see why we would expect it to fail, consider the halting set $K \subset \mathbb{N}$. If it were constructively true that

$$n \in K \vee n \notin K$$

(n free) then there would be an algorithm that, given any natural number n, told us whether or not n was in the halting set. Assuming Church's thesis, such an algorithm can't exist. ■

Example 3.1.2 The classical tautology

$$\neg\neg\phi \to \phi$$

is not constructively valid either, though it's harder to see why. Intuitively, $\neg\neg\phi$ means "assuming that ϕ is false leads to a contradiction"; this may not entail ϕ. For example, it's absurd to assume that Mr. X, your supervisor, doesn't exist; but that doesn't mean you have an effective procedure for finding him. ■

Exercise 3.1.3 Show that if we add the rule schema

$$\frac{}{\vdash \neg\neg\phi \to \phi}$$

then $\vdash \phi \vee \neg\phi$ is always derivable. This is perhaps a better reason for rejecting the schema. Would you expect

$$(\neg\neg\phi \to \phi) \to (\phi \vee \neg\phi)$$

to be constructively valid, in general? ■

Exercise 3.1.4 For any formula ϕ, give a constructive proof of

$$\vdash \neg\neg\neg\phi \rightarrow \neg\phi$$

Why is this reasonable? Well, assume that it's ridiculous to suppose that assuming ϕ is false leads to a contradiction; we want to show that to assume ϕ is absurd. But suppose ϕ holds. Then certainly assuming ϕ to be false would lead to a contradiction. But that's ridiculous. (Computational information doesn't enter into it.) ∎

I shouldn't carry on like this. To focus on counterexamples, and on the unfamiliar behaviour of constructive negation, misses the point of constructive logic: namely, that it is somehow meant to capture a subjective or algorithmic notion of proof. In any case, you don't need to be a constructivist to accept constructive logic as a formal tool. We can discuss the philosophy elsewhere.

The Heyting algebra interpretation

In fact, constructive logic turns out not to be so subjective after all: it turns out to describe many situations that arise quite naturally in classical mathematics. In order to give some examples of this (and because we need it later) we will describe an algebraic model for constructive propositional logic. To do this, we need to define lattices with enough structure to let us model all the propositional connectives. Note that the connectives are not interdefinable, as they are classically.

Definition 3.1.5 *A poset H is a **Heyting algebra** if we have elements*

$$\top \in H, \qquad \bot \in H$$

and operations

$$\wedge\colon H \times H \rightarrow H, \qquad \vee\colon H \times H \rightarrow H, \qquad \rightarrow\colon H \times H \rightarrow H$$

such that $\langle H, \wedge, \vee, \top, \bot\rangle$ is a lattice and

$$c \wedge a \leq b \quad \longleftrightarrow \quad c \leq a \rightarrow b$$

for all $a, b, c \in H$. (Sorry about the overloading of the symbol "$\rightarrow$".)

Note that a poset, though of as a category, is a Heyting algebra exactly when it is cartesian closed and has finite coproducts. That is, for all $a \in H$, the map $a \rightarrow (-)\colon H \rightarrow H$ is right adjoint to $(-) \wedge a$.

Exercise 3.1.6 Show that any Heyting algebra is a *distributive* lattice:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c), \qquad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

for all $a, b, c \in H$. ∎

Example 3.1.7 Any Boolean algebra is a Heyting algebra; put

$$a \rightarrow b \;=\; \neg a \vee b$$

For example, if X is an ordinary set, its powerset PX is a Heyting algebra. ∎

Example 3.1.8 Let X be a topological space. Then its lattice $\mathcal{O}(X)$ of open sets is a Heyting algebra; if $U, V \subseteq X$ are open sets, put

$$U \to V \ = \ \text{the interior of } (X - U) \cup V$$

However, $\mathcal{O}(X)$ is rarely a Boolean algebra. ■

Exercise 3.1.9 What are the open sets $U \subseteq \mathbb{R}$ satisfying $\neg\neg U = U$? ■

If H is a Heyting algebra, we can use it to give a semantics for constructive propositional logic; we think of the elements of H as 'truth-values'. Suppose A is the set of atomic propositions, and let $\nu: A \to H$ be a *valuation*, assigning a truth-value to each atomic proposition. Then we can extend ν to a map

$$[\![\cdot]\!]_\nu : \mathcal{L}_A \to H$$

assigning a truth-value to each formula $\phi \in \mathcal{L}_A$ as follows:

$$[\![a]\!]_\nu = \nu(a), \quad [\![\mathsf{true}]\!]_\nu = \top, \quad [\![\mathsf{false}]\!]_\nu = \bot$$

$$[\![\phi \wedge \psi]\!]_\nu = [\![\phi]\!]_\nu \wedge [\![\psi]\!]_\nu, \quad [\![\phi \vee \psi]\!]_\nu = [\![\phi]\!]_\nu \vee [\![\psi]\!]_\nu, \quad [\![\phi \to \psi]\!]_\nu = [\![\phi]\!]_\nu \to [\![\psi]\!]_\nu$$

Proposition 3.1.10 *If $\vec{\phi} \vdash \psi$ is derivable,*

$$[\![\phi_1]\!]_\nu \wedge \cdots \wedge [\![\phi_n]\!]_\nu \leq [\![\psi]\!]_\nu$$

for any valuation ν.

If $[\![\phi]\!]_\nu = \top$ for any valuation ν, we say that $H \models \phi$.

Proposition 3.1.11 *If $H \models \phi$ for all Heyting algebras H, $\vdash \phi$ is derivable.*

The proofs of these two propositions, *soundness* and *completeness*, are left to the reader; they are exactly the same as the ones for classical logic/Boolean algebras.

Predicate logic

How would we extend our system to a constructive version of predicate logic? Atomic propositions, and hence formulas, are now allowed to contain variables x; and we extend the syntax of formulas to allow universal and existential quantification:

$$\phi \quad ::= \quad \cdots \,|\, \forall x.\, \phi \,|\, \exists x.\, \phi$$

If a formula ϕ might contain a free variable x, we often write "$\phi(x)$" to emphasise this.

We could also imagine a many-sorted version. The internal language of a topos is many-sorted, of course; one only quantifies over the elements of a particular set. For the moment we'll just stick to a single-sorted theory.

A sequent now looks like

$$[\vec{x}]\ \vec{\phi} \vdash \psi$$

where $\vec{x} = x_1, \ldots, x_k$ is a list of variables, such that

$$FV(\vec{\phi}) \cup FV(\psi) \subseteq \{x_1, \ldots, x_k\}$$

We could again use the introduction and elimination rules of natural deduction (see the appendix). But instead, we'll present some equivalent rules which look more categorical. I learnt them from Andy Pitts.

$$\frac{[\vec{x}, y]\ \vec{\phi} \vdash \psi(y)}{[\vec{x}]\ \vec{\phi} \vdash \forall y.\, \psi(y)} \qquad \frac{[\vec{x}]\ \vec{\phi} \vdash \forall y.\, \psi(y)}{[\vec{x}, y]\ \vec{\phi} \vdash \psi(y)}$$

$$\frac{[\vec{x}, y]\ \vec{\phi}, \psi(y) \vdash \zeta}{[\vec{x}]\ \vec{\phi}, \exists y.\, \psi(y) \vdash \zeta} \qquad \frac{[\vec{x}]\ \vec{\phi}, \exists y.\, \psi(y) \vdash \zeta}{[\vec{x}, y]\ \vec{\phi}, \psi(y) \vdash \zeta}$$

(There are also trivial rules for manipulating the list $\vec{x}$, e.g. for permuting variables.)

If V is a set of variables, let

$$\mathcal{L}(V) = \{\text{formulas } \phi \text{ such that } FV(\phi) \subseteq V\}$$

Then if x is a fresh variable, there's an obvious inclusion

$$\mathcal{L}(V) \hookrightarrow \mathcal{L}(V \cup \{x\})$$

The rules suggest that $\forall x$ and $\exists x$ are right and left adjoint to this inclusion. Compare the remarks at the end of example 2.3.2.

The following example shows how we can formalise this intuition at the semantic level; it is the basis for our interpretation of quantification in the more general setting of toposes.

Example 3.1.12 If X is a set let $P(X)$ be the powerset of X; note that $P(X)$ is a complete lattice. One can regard a subset $X' \in P(X)$ as a predicate on X (the interpretation of a formula ϕ with a single free variable).

Let $2 = \{tt, f\!f\}$; we can identify 2 with $P(1)$. Then for any set X, the elements $tt, f\!f \in 2$ correspond to the subsets $X, \varnothing \subseteq X$; this gives us a map $2 \to X$. Conversely, given $X' \subseteq X$ we can define

$$\forall_X X' = \begin{cases} tt & \text{if } X' = X \\ f\!f & \text{otherwise} \end{cases}$$
$$\exists_X X' = \begin{cases} tt & \text{if } X' \neq \varnothing \\ f\!f & \text{otherwise} \end{cases}$$

This is the obvious semantics for $\forall x.\, \phi$, $\exists x.\, \phi$. It gives us maps $\forall_X, \exists_X : P(X) \to 2$ that are right and left adjoint to the map $2 \to P(X)$.

One can generalise this. Let X, Y be sets, and consider the projection function

$$\pi_0 : Y \times X \to Y$$

This gives rise to an 'inverse image' function

$$\pi_0^{-1} : P(Y) \to P(Y \times X)$$

sending $Y' \mapsto \pi_0^{-1} Y'$. This has right and left adjoints defined by

$$\forall_{\pi_0}, \exists_{\pi_0} : P(Y \times X) \to P(Y)$$

$$\begin{array}{rcl} Z \subseteq Y \times X & \mapsto & \{y \in Y \mid \langle y, x\rangle \in Z \text{ for all } x \in X\} \\ Z \subseteq Y \times X & \mapsto & \{y \in Y \mid \langle y, x\rangle \in Z \text{ for some } x \in X\} \end{array}$$

Identifying a subset of $Y \times X$ with a predicate $\phi(y, x)$, this suggests a categorical interpretation of $\forall x$ and $\exists x$. ■

Exercise 3.1.13 In the case $X = Y = \mathbb{R}$, give a geometrical interpretation of $\pi_0^{-1}, \forall_{\pi_0}, \exists_{\pi_0}$. ■

The verifications are an easy exercise. Incidentally, the idea that "quantification is adjoint to substitution" is due to Bill Lawvere; it is not special to topos theory, but is the basis for the whole of categorical logic, just about. Note that we are talking about a many-sorted logic here.

We are actually glossing over something here. The right and left adjoints $\forall_{\pi_0}, \exists_{\pi_0}$ do exist, but they also satisfy a certain 'compatibility' property called the *Beck-Chevalley condition.* From a logical point of view this condition is rather trivial, but categorically, it must be stated explicitly[2]. We will say exactly what it is later (see proposition 3.4.30).

Also, observe that to make sense of a formula ϕ containing free variables (possibly of different sorts, if our logic is many-sorted), we need to interpret it using a 'parameter set'—here, the set $X \times Y$. That is, in a set-theoretic interpretation, lists of free variables $x_1, \ldots, x_n$ ('environments' or 'contexts') correspond to parameter sets $X_1 \times \cdots \times X_n$; and quantification involves changing the parameter set, by dropping a parameter—here, the parameter set X. There is a neat categorical way of handling the bookkeeping related to environments, namely the systematic use of slice categories. More on this later.

Some categorical definitions

Finally, we need to list two definitions used later on.

Definition 3.1.14 *Let $\mathcal{E}$ be a category with pullbacks and let $f: X \to Y$ be an arrow in $\mathcal{E}$. The* ***kernel pair*** *$E \rightrightarrows X$ of f is the pair of arrows p_0, p_1 appearing in the pullback square*

$$\begin{tikzcd} E \arrow[r, "p_1"] \arrow[d, "p_0"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

Intuitively,

$$E = \{\langle x, x' \rangle \mid f(x) = f(x')\}$$

and p_0, p_1 are the two projections. By the way, whenever we say that "intuitively" something has a set-theoretic meaning, in a topos we can make this rigorous using the internal language—so such statements aren't just rubbish. However, *at the moment* they are only meant as guides.

Definition 3.1.15 *Let $\mathcal{E}$ be a category, X an object of $\mathcal{E}$. If $m_1: X_1 \rightarrowtail X$ and $m_2: X_2 \rightarrowtail X$ are monos, we say that*

$$\begin{array}{ll} m_1 \leq m_2 & \textit{if } m_1 \textit{ factors through } m_2 \\ m_1 \approx m_2 & \textit{if } m_1 \leq m_2 \textit{ and } m_2 \leq m_1 \end{array}$$

A ***subobject*** *of X is a $\approx$-equivalence class of monos into X. The subobjects of X form a poset Sub(X), which is equivalent to the full subcategory of $\mathcal{E}/X$ whose objects are the monos $\left(\begin{array}{c} X' \\ \downarrow m \\ X \end{array}\right)$.*

[2]There are examples in which the adjoints exist but the Beck-Chevalley condition fails.

It is usual to blur the distinction between monos and subobjects. Note that $\mathrm{Sub}(X)$ is always a poset with a top element, namely X itself (i.e. the subobject having the identity map as a representative). Clearly

- if $\mathcal{E}$ has an initial object, then $\mathrm{Sub}(X)$ has a bottom element; and
- if $\mathcal{E}$ has pullbacks, then $\mathrm{Sub}(X)$ has meets.

It turns out that if $\mathcal{E}$ is a topos, $\mathrm{Sub}(X)$ is a Heyting algebra; so, thinking of subobjects of X as 'predicates' on X, this allows us to give an interpretation of constructive propositional logic. The quantifiers are handled in the way suggested earlier.

3.2 Toposes

Toposes were invented by Grothendieck in the early 1960s; however, modern topos theory really begins with the remarkable work of Lawvere and Tierney in 1969–70, and the present 'elementary' definition of a topos is due to them (as well as most of the unattributed results in the next three sections). Lawvere was certainly the first person to see that a topos could be regarded as a—constructive—universe of sets, and to exploit this. But it's important to remember that toposes were, and still are, used in many different ways. Depending on the circumstances, a topos can be thought of as

- a model of constructive set theory;
- a (special kind of) first-order theory;
- a (generalised) topological space; or
- a category of generalised spaces.

Moreover, at least in more advanced developments, these four different viewpoints interact very closely; a single proof may flick from one to the other and back again. But in these notes, we'll only be concerned with the first interpretation: that of a topos as a universe of sets[3].

[3] I might mention one respect in which these notes differ from the accounts of (for example) Lambek/Scott or Bell. A logician might write down a formal language for set theory and then view toposes as models for that language: that is, traditional set theories or type theories are taken to be the fundamental way of giving concrete expression to our set-theoretic intuitions; hence set theory is an essential mediator between these informal intuitions and the categorical models provided by topos theory, and cannot be bypassed. A category theorist might take a different view: that topos theory is itself a *direct* way of formulating our informal notion of 'set' mathematically—that it is an alternative to traditional logic. In this view, the relationship between topos theory and 'old-fashioned' set theory is a partnership of equals, and we must not be distracted by it from the autonomous development of topos theory as a branch of category theory and from the effort to find a new, categorical expression of our informal conception of logic and set theory; to rely too exclusively on the old set-theoretic tools would be to risk missing the insights that the categorical approach provides, such as Lawvere's realisation that quantifiers are adjoints, or the 'behavioural' characterisations of set-theoretic constructions in terms of universal properties rather than particular representations. But having said this, I should admit that I have not in these notes taken the most radical stance possible for a category theorist; indeed, once we have gained a certain amount of categorical insight, in theoretical computing (in contrast to some of the other branches of pure or applied mathematics in which topos theory has been found to be a useful tool) it is in fact often more convenient to talk about sets and elements than about objects and arrows, especially since the former way of working is by far the more familiar to most computer scientists.

Note: A lot of the proofs are very long. As this is only an outline, I've chosen some representative arguments and tried to spell them out in fairly long-winded detail. The other proofs are omitted entirely—they appear in the references, though.

Definition 3.2.1 *A category $\mathcal{E}$ is a **topos** if*

1. *it has finite limits;*
2. *it has exponentials (i.e. is cartesian closed); and*
3. *it has a subobject classifier—see below.*

Such a category $\mathcal{E}$ is often called an *elementary topos* in the literature; this is because there was an earlier, more complicated definition of 'topos' used in the Grothendieck school of algebraic geometry[4].

Definition 3.2.2 *Let $\mathcal{E}$ be a category with a terminal object; then a **subobject classifier** in $\mathcal{E}$ is a map $\top: 1 \to \Omega$ with the following property: if $m: X' \rightarrowtail X$ is any mono in $\mathcal{E}$, there is a unique map $\chi_m: X \to \Omega$ (the **classifying map** of $m: X' \rightarrowtail X$) such that the following square is a pullback.*

$$\begin{tikzcd}
X' \arrow[r, "m"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "\chi_m"] \\
1 \arrow[r, "\top"'] & \Omega
\end{tikzcd}$$

Exercise 3.2.3 A category $\mathcal{E}$ with finite limits has a subobject classifier iff the functor

$$\mathcal{E}^op \to \mathcal{S}ets \qquad X \mapsto \mathrm{Sub}(X)$$

is representable. ■

Note that a subobject classifier is unique up to isomorphism, like any other entity characterised by a universal property; also note that $\top$ is mono—it's a 'generic subobject'. We usually refer to Ω as 'the subobject classifier', though the map $\top$ is really part of the data. Similarly, we often write $\chi_{X'}$ instead of χ_m if the mono m is understood.

Having a subobject classifier around can be thought of as a kind of *separation* axiom: there is a bijection between 'subsets' and 'predicates'. Anyway, the motivation for the definition of topos is the following example.

[4]It's perhaps worth noting here that the theory of elementary toposes was not the first categorical axiomatisation of set theory: indeed, Lawvere's thesis set out an elementary theory of the category of sets. There are three reasons why that formulation gave way to topos theory: the fact that the axioms of a topos are so much simpler and more elegant; the large number of toposes that occur in mathematics, which is a consequence of the greater flexibility of the axioms; and the many-sided nature of topos theory, in particular its relationships with geometry.

Example 3.2.4 $\mathcal{S}ets$, the ordinary category of sets, is a topos. Finite limits can be calculated in the usual way; exponentials are set-theoretic function spaces; and the subobject classifier is $2 = \{tt, ff\}$ (so that if $X' \subseteq X$, $\chi_{X'}$ is just the characteristic function of X'). ∎

Example 3.2.5 The category $\mathcal{S}ets_f$ of finite sets is also a topos. ∎

Example 3.2.6 $\mathcal{S}ets^2$, the category whose objects are pairs of sets and whose arrows are pairs of functions, is a topos; finite limits and exponentials are calculated componentwise, and the subobject classifier is $\langle 2, 2\rangle$. More generally, if I is a set then the category $\mathcal{S}ets^I$ of I-indexed families of sets is a topos, for the same reasons. ∎

Example 3.2.7 Let $\mathcal{S}ets^{\rightarrow}$ be the following category:

objects: functions $f: X \to Y$ where X and Y are sets

arrows: a morphism from f' to f is given by a commutative square

$$\begin{tikzcd} X' \arrow[r, "f'"] \arrow[d, "g_1"'] & Y' \arrow[d, "g_2"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

Then $\mathcal{S}ets^{\rightarrow}$ is a topos; an object of $\mathcal{S}ets^{\rightarrow}$ can be though of as 'one set evolving into another'.

Finite limits are again calculated componentwise. To describe the subobject classifier, first note that a morphism in $\mathcal{S}ets^{\rightarrow}$ is mono exactly when g_1, g_2 are mono; so given a mono, we may as well assume that $X' \subseteq X$ and $Y' \subseteq Y$. We let Ω be the object

$$\{tt, b, ff\} \xrightarrow{e} \{tt, ff\}$$

where $tt, b \mapsto tt$ and $ff \mapsto ff$ (so b stands for 'will become true'). The classifying map of a subobject $\langle X', Y'\rangle \rightarrowtail \langle X, Y\rangle$ is given by

$$\begin{array}{rcccl} X & \longrightarrow & \{tt, b, ff\} & x \mapsto & \begin{cases} tt & \text{if } x \in X' \\ b & \text{if } x \notin X' \text{ but } f(x) \in Y' \\ ff & \text{otherwise} \end{cases} \\ Y & \longrightarrow & \{tt, ff\} & y \mapsto & \begin{cases} tt & \text{if } y \in Y' \\ ff & \text{if } y \notin Y' \end{cases} \end{array}$$

Exponentials also have a 'two-stage' description. Given objects $f: X \to Y$, $f': X' \to Y'$, the exponential $(X' \xrightarrow{f'} Y')^{(X \xrightarrow{f} Y)}$ is given by $Z_1 \to Z_2$, where

$$\begin{array}{rcl} Z_1 & = & \{g_1: X \to X', g_2: Y \to Y' \mid f'g_1 = g_2 f\} \\ Z_2 & = & \{g_2: Y \to Y'\} \end{array}$$

That is, Z_1 is the set of all morphisms $f \to f'$ in $\mathcal{S}ets^{\rightarrow}$, whereas Z_2 is just the set of functions $Y \to Y'$. ∎

Example 3.2.8 Let $\mathcal{S}ets^{\omega}$ be the category whose objects are sequences of functions

$$X_0 \xrightarrow{f_1} X_1 \xrightarrow{f_2} X_2 \xrightarrow{f_3} X_3 \xrightarrow{f_4} \cdots$$

where $X_0, X_1, X_2 \dots$ are sets; an object of $\mathcal{S}ets^{\omega}$ can be though of as a 'set evolving through discrete time'. Then $\mathcal{S}ets^{\omega}$ is also a topos.

Finite limits are calculated as before. The subobject classifier is

$$\Omega_0 \xrightarrow{n_1} \Omega_1 \xrightarrow{n_2} \Omega_2 \xrightarrow{n_3} \Omega_3 \xrightarrow{n_4} \cdots$$

where

$$\begin{array}{rcl} \Omega_0 & = & \{tt, b_1, b_2, b_3, b_4 \dots, ff\} \\ \Omega_1 & = & \{tt, b_2, b_3, b_4 \dots, ff\} \\ \Omega_2 & = & \{tt, b_3, b_4 \dots, ff\} \\ & \vdots & \end{array}$$

and $n_i(ff) = ff$, $n_i(tt) = n_i(b_i) = tt$, $n_i(b_j) = b_j$ $(j > i)$. In this case, b_i can be interpreted as the truth-value 'becoming true at stage i'.

Given $X_0 \to X_1 \to X_2 \to \cdots$ and $Y_0 \to Y_1 \to Y_2 \to \cdots$, their exponential is

$$Z_0 \longrightarrow Z_1 \longrightarrow Z_2 \longrightarrow Z_3 \longrightarrow \cdots$$

where

$$\begin{array}{rcl} Z_0 & = & \{\text{compatible families } g_i\colon X_i \to Y_i \ (i \geq 0)\} \\ Z_1 & = & \{\text{compatible families } g_i\colon X_i \to Y_i \ (i \geq 1)\} \\ Z_2 & = & \{\text{compatible families } g_i\colon X_i \to Y_i \ (i \geq 2)\} \\ \vdots & & \end{array}$$

■

Exercise 3.2.9 Let P be any partial order. Show that $\mathcal{S}ets^{P}$ is a topos. ■

Exercise 3.2.10 Let $\mathcal{S}ets^{\rightrightarrows}$ be the category whose objects are pairs of functions

$$d_0, d_1\colon E \rightrightarrows V$$

where E, V are sets. Explain why $\mathcal{S}ets^{\rightrightarrows}$ can be regarded as the category of directed graphs (possibly with loops). Show that it's a topos. ■

Exercise 3.2.11 Show that the toposes just described all have finite colimits. ■

Exercise 3.2.12 (Harder.) Let G be a group. Then the category $\mathcal{S}ets^{G}$ of G-sets is a topos. ■

Exercise 3.2.13 (Hardest.) Let $\mathbb{C}$ be a small category. Then the functor category $\mathcal{S}ets^{\mathbb{C}}$ is a topos. See [12, 1.12]. ■

Another example, a fundamental one, is the following: if X is a topological space, then the category $\mathrm{Sh}(X)$ of *sheaves* on X is a topos. In fact, the notion of (Grothendieck) topos arose out of attempts to define a notion of 'topology' and 'sheaf' over a category, generalising from (the lattice of opens of) a topological space. However, you don't need to understand sheaves to understand these notes.

Functors between toposes

Definition 3.2.14 *Let $\mathcal{E}$ and $\mathcal{F}$ be toposes. A functor $F: \mathcal{E} \to \mathcal{F}$ is a **logical functor** if it preserves finite limits, exponentials and the subobject classifier (up to isomorphism).*

Example 3.2.15 The inclusion functor $\mathcal{S}ets_f \hookrightarrow \mathcal{S}ets$ is logical, and so are the projection functors $\mathcal{S}ets \times \mathcal{S}ets \rightrightarrows \mathcal{S}ets$. ■

We will shortly see that any topos has finite colimits, and that any logical functor preserves them. It is often useful to impose an extra condition which, for example, will ensure that the functor preserves *any* colimits that happen to exist.

Definition 3.2.16 *A logical functor is **atomic** if it has a right adjoint.*

Exercise 3.2.17 Which of the logical functors just mentioned are atomic? ■

Logical functors suffice for our purposes. However, it turns out that this is not, in general, the most useful notion of morphism between toposes. If we are thinking of a topos as a theory, or a space, or a category of spaces, it is the so-called *geometric morphisms* that are important. However, we won't need them, and the definition looks quite odd at first, so they won't be discussed here.

Power objects

In ordinary set theory we usually start with pairing, powersets, a separation axiom and some other axioms and construct everything else from those. It turns out that any object X in a topos has a 'powerset' PX, and that PX can in fact be characterised by a universal property.

Definition 3.2.18 *Let $\mathcal{E}$ be a category with finite limits, and let $X \in \mathcal{E}$. A **power object** for X is an object PX together with a mono $\in_X \rightarrowtail X \times PX$, the **membership relation**, with the following universal property: given any object Y and any relation $R \rightarrowtail X \times Y$, there is a unique arrow $\downarrow_R: Y \to PX$ making the following square a pullback:*

$$\begin{tikzcd} R \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \times Y \arrow[d, "1_X \times \downarrow_R"] \\ \in_X \arrow[r] & X \times PX \end{tikzcd}$$

Example 3.2.19 Let's see what this means in $\mathcal{S}ets$. For a given set X, $\in_X$ is the relation

$$\{\langle x, X' \rangle \mid x \in X'\}$$

And if $R \subseteq X \times Y$, the map $\downarrow_R: Y \to PX$ takes

$$y \mapsto \{x \mid \langle x, y \rangle \in R\}$$

■

Proposition 3.2.20 *If $\mathcal{E}$ is a topos, any object $X \in \mathcal{E}$ has a power object.*

Proof. Let $PX = \Omega^X$, and let $\in_X \rightarrowtail X \times PX$ be (a representative for) the subobject classified by the evaluation map $ev: X \times \Omega^X \to \Omega$. If $R \rightarrowtail X \times Y$, $\downarrow_R: Y \to PX$ is the exponential transpose of its classifying map $\chi_R: X \times Y \to \Omega$. ■

It turns out that having powersets is extremely useful—in particular, we can perform many other constructions on 'sets' using powersets. The first example, like most of the others, should already be familiar from set theory.

Theorem 3.2.21 (Kock) *If a category $\mathcal{E}$ has finite limits and power objects, it is a topos.*

Proof. Intuitively, Y^X ought to be a subobject of $P(X \times Y)$:

$$Y^X = \{R \subseteq X \times Y \mid R \text{ is a functional relation}\}$$

One can find a categorical version of this definition and prove that it indeed defines the categorical exponential. The details are rather tricky, though. ■

That is, one can define exponentials in terms of finite limits and power objects. In fact, a topos is often *defined* to be a category with finite limits and power objects. But as exponentials are so useful, and as the proof of the proposition is quite tricky, our original definition seems more convenient.

(Interestingly, if we want to develop a 2-categorical version of topos theory, the 'power object' definition seems more suitable: in this case PX classifies *profunctors* rather than relations. However, this theory is still not very well developed.)

3.3 Categorical properties

The axioms for a topos are quite simple. However, they have surprisingly strong consequences—as they must, if a topos is to give us an interpretation for set theory. In this section, we will see a few glimpses of the remarkably rich structure that every topos has. Not many proofs are given; they are mostly long, and difficult, diagram chases. There's no doubt that the pioneering topos theorists of the early 1970s all drank Carling Black Label. Anyway, we'll start with something easy (but important).

Proposition 3.3.1 *A topos is* ***balanced:*** *i.e. any map $f: X \to Y$ in $\mathcal{E}$ which is both epi and mono is an isomorphism.*

Proof. If f is mono, the definition of subobject classifier tells us that it's the equalizer of the maps $\top, \chi_f: Y \to \Omega$, where $\top$ is the composite

$$Y \longrightarrow 1 \xrightarrow{\top} \Omega$$

But if a map is both an equalizer and and epi, then an easy diagram chase shows that it's an isomorphism. ■

Example 3.3.2 Note that not all categories are balanced. For example, in the category of rings, the inclusion $\mathbb{Z} \hookrightarrow \mathbb{Q}$ is both mono and epi, but it's certainly not an isomorphism. And in the category of topological spaces, any dense inclusion is both mono and epi. ■

Epis in these categories somehow do not behave like *surjections*, whereas epis in a topos do; but we won't try and make this vague intuition precise here.

Partial map classifiers

Because a topos is cartesian closed, it provides a convenient setting for working with total functions between 'sets'. But in computer science as well as mathematics, it's useful to be able to talk about partial functions as well.

Definition 3.3.3 *A* ***partial map*** $f: X \rightharpoonup Y$ *in a category consists of a mono* $X' \rightarrowtail X$ *together with a map* $f: X' \to Y$.

Actually, we should identify two partial maps if there is an isomorphism $X' \cong X''$ between their codomains making the appropriate triangles commute; cf. the definition of 'subobject' in a category. But, as for subobjects, we will instead speak loosely most of the time, and identify a partial map with a particular representative for it.

Definition 3.3.4 *Let $\mathcal{E}$ have finite limits, and let $Y \in \mathcal{E}$. Then a* ***partial map classifier*** *for Y is an object $\tilde{Y}$ together with a mono $Y \rightarrowtail \tilde{X}$, such that given any partial map $f: X \rightharpoonup Y$, there is a unique map $\tilde{f}: X \to \tilde{Y}$ making the following square a pullback:*

$$\begin{tikzcd} X' \arrow[r, "f"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d] \\ X \arrow[r, "\tilde{f}"'] & \tilde{Y} \end{tikzcd}$$

$\tilde{f}$ is said to ***classify*** *f.*

Note that any map $g: X \to \tilde{Y}$ gives us a partial map $X \rightharpoonup Y$: form the pullback of g along $Y \rightarrowtail \tilde{Y}$.

Example 3.3.5 As usual, it's useful to see what the definition means in $\mathcal{S}ets$. In this case we put $\tilde{Y} = Y \dot{\cup} \{*\}$, where '$*$' stands for 'undefined'. Given a partial map $f: X \rightharpoonup Y$, we define $\tilde{f}: X \to Y \dot{\cup} \{*\}$ by

$$\tilde{f}(x) = \begin{cases} f(x) & x \in X' \\ * & x \notin X' \end{cases}$$

Alternatively, we can identify $\tilde{Y}$ with the set of subsets of Y containing at most one element; then

$$\tilde{f}(x) = \{f(x') |\, x' \in X', x' = x\}$$

(This slightly eccentric viewpoint is the one that generalises.) ■

Exercise 3.3.6 Describe partial map classifiers in $\mathcal{S}ets^{\to}$. ■

The following result, which goes back to the earliest work of Lawvere-Tierney on elementary toposes, is an important technical lemma as well as being of great interest in its own right.

Theorem 3.3.7 *Any object Y in a topos has a partial map classifier.*

Proof. We will give the construction, which is quite natural; it is written out in more 'set-theoretic' language in section 3.7. The verification is rather complicated, and will be omitted.

Let $\chi_\delta\colon Y \times Y \to \Omega$ classify the diagonal map $Y \overset{\delta}{\rightarrowtail} Y \times Y$, and let $\{\cdot\}\colon Y \to \Omega^Y$ be the exponential transpose of χ_δ. Intuitively,

$$\chi_\delta(y, y') = |y = y'| \qquad \text{and} \qquad \{\cdot\}(y) = \{y' |\, y' = y\} = \{y\}$$

In fact, $\{\cdot\}$ is often called the **singleton map**.

Let $\phi\colon \Omega^Y \times Y \to \Omega$ classify the mono $\langle\{\cdot\}, 1_Y\rangle\colon Y \rightarrowtail \Omega^Y \times Y$, and let $\bar{\phi}\colon \Omega^Y \to \Omega^Y$ be its exponential transpose. Intuitively, if $Y' \in PY = \Omega^Y$,

$$\bar{\phi}(Y') = \{y |\, \{y\} = Y'\}$$

We obtain $\tilde{Y}$ as an equalizer:

$$\begin{tikzcd} \tilde{Y} \arrow[r] & \Omega^Y \arrow[r, shift left, "\bar{\phi}"] \arrow[r, shift right, "1_{\Omega^Y}"'] & \Omega^Y \end{tikzcd}$$

So intuitively,

$$\tilde{Y} = \{Y' \subseteq Y |\, Y' = \{y |\, \{y\} = Y'\}\}$$

That is, $\tilde{Y}$ is the set of subsets of Y with at most one element (for if $Y' \in \tilde{Y}$ then if $x, x' \in X'$, $\{x\} = X' = \{x'\}$ and so $x = x'$). We can check that $\{\cdot\}\colon Y \rightarrowtail \Omega^Y$ equalizes $\bar{\phi}$ and 1_{Ω^Y}, so we do get a map $Y \to \tilde{Y}$ as desired, and it's mono.

Given a partial map $f\colon X \rightharpoonup Y$, its graph $X' \rightarrowtail X \times Y$ is classified by a map $X \times Y \to \Omega$; one can show that its exponential transpose $X \to PY$ factors through $\tilde{Y}$, giving us the map $\tilde{f}$. ■

(See also 3.7.1.)

In fact, the notion of partial map classifier predates the notion of subobject classifier, and seems to have motivated it. In particular, note that we must have $\Omega \cong \tilde{1}$.

As well as being interesting in its own right, the existence of partial map classifiers is a useful technical tool. Here is one example; we'll see another in the next section.

Exercise 3.3.8 Suppose we have a pushout square

$$\begin{tikzcd} X' \arrow[r] \arrow[d, "m"'] & Y \arrow[d, "n"] \\ X \arrow[r] & Z \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

and m is mono. Then n is mono and the square is a pullback. (Hint: let $h\colon X \to \tilde{Y}$ represent the partial map $(X' \rightarrowtail X, X' \to Y)$; then

$$Y \longrightarrow \tilde{Y} \xleftarrow{h} X$$

factors through the pushout square.) ■

Finite colimits

Every topos has them. This fact is of central importance. If it were not true, we would add it as an axiom; in fact, finite cocompleteness was one of the original axioms proposed by Lawvere and Tierney. But, magically, it actually follows from the axioms we already have.

Theorem 3.3.9 (Mikkelsen) *Every topos $\mathcal{E}$ has finite colimits.*

Proof. Mikkelsen's original proof was long; we will outline an elegant argument due to Paré, without going into the details. Actually, we won't need to know the proof in the sequel, just the theorem; so readers who are not familiar with monads can skip over the argument.

Recall that for each $X \in \mathcal{E}$ we had a power object $PX = \Omega^X$. If $f: X \to Y$, we can define a map $Pf: PY \to PX$: it's the exponential transpose of

$$\Omega^Y \times X \xrightarrow{1_{\Omega^Y} \times f} \Omega^Y \times Y \xrightarrow{ev} \Omega$$

Intuitively, Pf sends a subset $Y' \subseteq Y$ to its inverse image $f^{-1}Y' \subseteq X$.

We have defined a functor $P: \mathcal{E}^{op} \to \mathcal{E}$. In fact, this also defines a functor $P_*: \mathcal{E} \to \mathcal{E}^{op}$; and it's easy to see that $P_* \dashv P$. Paré showed, using Beck's theorem, that the functor P is monadic: that is, that $\mathcal{E}^{op}$ is the category of algebras for the monad induced by this adjunction.

But a monadic functor creates limits; so since $\mathcal{E}$ has finite limits, $\mathcal{E}^{op}$ has finite limits. In other words, $\mathcal{E}$ has finite colimits. ∎

Exercise 3.3.10 Here is a construction which is needed in Paré's proof. Let $m: X \to Y$ be mono, and let $\exists_m: PX \to PY$ be the classifying map of the mono

$$\in_X \rightarrowtail PX \times X \xrightarrow{1_{PX} \times m} PX \times Y$$

Show that $Pm.\exists_m = 1_{PX}$. What, intuitively, does $\exists_m$ do? ∎

There is an intuition behind this proof. A category of algebras for a monad ought to be some category of 'algebraic structures' of some sort. In the case $\mathcal{E} = \mathcal{S}ets$, it was long known that $\mathcal{S}ets^{op}$ was the category of 'complete atomic Boolean algebras'; it was known to be monadic, since such things can be defined in terms of operations and equations. This was one of the starting-points of Mikkelsen's thesis: he showed that for any topos $\mathcal{E}$, the category of "complete atomic Boolean algebras in $\mathcal{E}$" was equivalent to $\mathcal{E}^{op}$.

Example 3.3.11 At first glance, the proof looks a bit abstract. To illustrate it, let's show that any topos $\mathcal{E}$ has an initial object 0. We'll only give the construction of 0, and construct the mediating map $0 \to X$ for any $X \in \mathcal{E}$. Note that this is already quite complicated! So it's lucky we have a theorem to do the work for us. In fact, we will never need this explicit description, which is given here only out of interest.

If $X \in \mathcal{E}$ let $!_X: X \to 1$ denote the unique map, and let $\alpha_X: X \to PPX = \Omega^{\Omega^X}$ be the exponential transpose of the evaluation map $\Omega^X \times X \to \Omega$.

Form the equalizer

$$E \longrightarrow \Omega \underset{\alpha_\Omega}{\overset{P!_{P\Omega}}{\rightrightarrows}} PP\Omega$$

Intuitively,

$$E = \{p \in \Omega \,|\, \forall f \in \Omega^{\Omega}.\, p \leftrightarrow f(p)\}$$

So we would expect E to be the empty set (consider the constant maps f).

Let $X \in \mathcal{E}$; we need to construct an arrow $E \to X$. Now since $\mathcal{E}^{op}$ is monadic over $\mathcal{E}$, every PP-algebra (i.e. every object $X \in \mathcal{E}^{op}$) has a canonical presentation as a coequalizer of a diagram of free PP-algebras. This means that we have an equalizer diagram

$$\begin{tikzcd} X \arrow[r] & PPX \arrow[r, shift left, "PP\alpha_X"] \arrow[r, shift right, "\alpha_{PPX}"'] & PPPPX \end{tikzcd}$$

Consider the diagram

$$\begin{tikzcd} E \arrow[r] \arrow[d, dotted] & \Omega \arrow[r, shift left, "P!_{P\Omega}"] \arrow[r, shift right, "\alpha_\Omega"'] \arrow[d, "P!_{PX}"'] & PP\Omega \arrow[d, "PPP!_{PX}"] \\ X \arrow[r] & PPX \arrow[r, shift left, "PP\alpha_X"] \arrow[r, shift right, "\alpha_{PPX}"'] & PPPPX \end{tikzcd}$$

The right-hand squares commute, so there is a unique map $E \to X$ making the left-hand square commute. ■

Paré's argument has a useful corollary which we'll need very soon.

Proposition 3.3.12 *Let $F\colon \mathcal{E} \to \mathcal{F}$ be a logical functor. Then F preserves finite colimits. Furthermore, if F has a left adjoint then it also has a right adjoint.*

Proof. Since F is logical, the following square commutes up to isomorphism:

$$\begin{tikzcd} \mathcal{E}^{op} \arrow[r, "F^{op}"] \arrow[d] & \mathcal{F}^{op} \arrow[d] \\ \mathcal{E} \arrow[r, "F"'] & \mathcal{F} \end{tikzcd}$$

Since F preserves finite limits, F^{op} preserves finite limits; that is, F preserves finite colimits. Furthermore, since $\mathcal{E}^{op}$ has reflexive coequalizers, a standard adjoint lifting theorem tells us that a left adjoint for F will lift to a left adjoint for F^{op}, i.e. a right adjoint for F. ■

3.4 Slice toposes

So far we have worked entirely within a single topos $\mathcal{E}$. It turns out that this is not enough. For example, if X is a set (an object of $\mathcal{E}$) we may want to talk about constructions 'parametrised' by X, or to interpret formulas $\phi(x)$ with a free variable $x : X$. The cleanest way to do this is to work inside the *slice topos* $\mathcal{E}/X$. The various slice toposes of $\mathcal{E}$ interact in a very nice way; it is this that lets us interpret indexed products and coproducts, and universal and existential quantification.

This section contains a few technical proofs—they're meant to get across some of the flavour of (the early days of) topos theory. However, you don't need to understand the proofs to be able to follow the rest of these notes; it's only necessary to understand the theorems.

Definition 3.4.1 *Let $\mathcal{E}$ be a topos and let $X \in \mathcal{E}$. The* ***slice category*** *$\mathcal{E}/X$ is defined as follows (cf. 2.2.9):*

objects: *maps* $\left(\begin{smallmatrix} Z \\ \downarrow h \\ X \end{smallmatrix}\right)$ *in* $\mathcal{E}$

arrows: *an arrow* $\left(\begin{smallmatrix} Z \\ \downarrow h \\ X \end{smallmatrix}\right) \to \left(\begin{smallmatrix} Z' \\ \downarrow h' \\ X \end{smallmatrix}\right)$ *is given by a map* $g\colon Z \to Z'$ *such that*

$$\begin{tikzcd} Z \arrow[rr, "g"] \arrow[dr, "h"'] & & Z' \arrow[dl, "h'"] \\ & X & \end{tikzcd}$$

Loosely, we say that the objects of $\mathcal{E}/X$ are maps into X, and the arrows are commutative triangles.

Here is the motivating example.

Example 3.4.2 $\mathcal{S}ets/X$ is equivalent to $\mathcal{S}ets^X$, the topos of X-indexed families of sets. For given an object $\left(\begin{smallmatrix} Z \\ \downarrow h \\ X \end{smallmatrix}\right) \in \mathcal{S}ets/X$ we get an X-indexed family of sets $\{Z_x\}_{x \in X}$ by letting

$$Z_x = h^{-1}\{x\}$$

Conversely, given any family $\{Z_x\}_{x \in X}$, we can get an object $\left(\begin{smallmatrix} Z \\ \downarrow \\ X \end{smallmatrix}\right) \in \mathcal{S}ets/X$ by forming the disjoint union

$$Z = \coprod_{x \in X} Z_x$$

These two constructions are mutually inverse. ■

If $\mathcal{E}$ is a general topos and $X \in \mathcal{E}$, it is not clear what '$\mathcal{E}^X$' might mean. But the example suggests that we may regard an object $\left(\begin{smallmatrix} Z \\ \downarrow \\ X \end{smallmatrix}\right) \in \mathcal{E}/X$ as an 'X-indexed family of objects of $\mathcal{E}$'; in other words, $\mathcal{E}/X$ can serve as a replacement for $\mathcal{E}^X$. When we come to define the internal language of $\mathcal{E}$, a set-theoretic formula with a free variable of type X will be interpreted, not in $\mathcal{E}$, but in $\mathcal{E}/X$: the slice topos captures the idea of an object, an arrow or a predicate 'varying over X'. This is, in fact, the whole reason for talking about slice categories in the first place. Here is the theorem; some further intuition is given in exercise 3.4.5.

Theorem 3.4.3 *Let $\mathcal{E}$ be a topos, and let $X \in \mathcal{E}$. Then*

- *$\mathcal{E}/X$ is a topos;*
- *the functor $X^*: \mathcal{E} \to \mathcal{E}/X$ sending*

$$Y \mapsto \begin{pmatrix} X \times Y \\ \downarrow \pi_0 \\ X \end{pmatrix}$$

  is logical; and
- *X^* has left and right adjoints $\Sigma_X, \Pi_X: \mathcal{E}/X \to \mathcal{E}$.*

Proof. Products in $\mathcal{E}/X$ are just pullbacks in $\mathcal{E}$, and equalizers are the same; also, monos in $\mathcal{E}/X$ are the same as monos in $\mathcal{E}$, so $X^*\Omega$ is a subobject classifier in $\mathcal{E}/X$.

X^* has a left adjoint Σ_X that sends $\begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix} \mapsto Z$. It follows that it preserves finite limits. It's also easy to check that it preserves exponentials. It remains to show that $\mathcal{E}/X$ has all exponentials, and hence is a topos; then, since X^* is a logical functor, 3.3.12 tells us that it also has a right adjoint Π_X.

The construction of general exponentials in $\mathcal{E}/X$ is a little involved, and may be skipped. But if you want to see it: suppose $\begin{pmatrix} Y \\ \downarrow f \\ X \end{pmatrix}, \begin{pmatrix} Z \\ \downarrow g \\ X \end{pmatrix} \in \mathcal{E}/X$. The pair

$$Y \overset{\langle f, 1_Y \rangle}{\rightarrowtail} X \times Y, \quad Y \xrightarrow{f} X$$

defines a partial map $\phi: X \times Y \rightharpoonup X$; so we have $\theta = \tilde{\phi}: X \times Y \to \tilde{X}$. Let $\bar{\theta}: X \to \tilde{X}^Y$ be its exponential transpose.

The exponential $\begin{pmatrix} E \\ \downarrow \\ X \end{pmatrix} = \begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix}^{\begin{pmatrix} Y \\ \downarrow \\ X \end{pmatrix}}$ is defined by the pullback

$$\begin{tikzcd} E \arrow[r] \arrow[d, "h"'] \arrow[dr, phantom, "\lrcorner", very near start] & \tilde{Z}^Y \arrow[d, "\tilde{g}^Y"] \\ X \arrow[r, "\bar{\theta}"'] & \tilde{X}^Y \end{tikzcd}$$

We must show that $\begin{pmatrix} E \\ \downarrow \\ X \end{pmatrix}$ has the correct universal property. Suppose $\begin{pmatrix} T \\ \downarrow h \\ X \end{pmatrix} \in \mathcal{E}/X$; we have to establish a natural bijection

$$\frac{\begin{pmatrix} T \\ \downarrow \\ X \end{pmatrix} \times \begin{pmatrix} Y \\ \downarrow \\ X \end{pmatrix} \to \begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix}}{\begin{pmatrix} T \\ \downarrow \\ X \end{pmatrix} \to \begin{pmatrix} E \\ \downarrow \\ X \end{pmatrix}}$$

where $\begin{pmatrix} T \\ \downarrow \\ X \end{pmatrix} \times \begin{pmatrix} Y \\ \downarrow \\ X \end{pmatrix}$ denotes the product in $\mathcal{E}/X$. Recall that it is calculated as the pullback of $h\colon T \to X$ and $f\colon Y \to X$; that is,

$$\begin{pmatrix} T \\ \downarrow \\ X \end{pmatrix} \times \begin{pmatrix} Y \\ \downarrow \\ X \end{pmatrix} = \begin{pmatrix} T \times_X Y \\ \downarrow \\ X \end{pmatrix}$$

Now, by the definition of $\begin{pmatrix} E \\ \downarrow \\ X \end{pmatrix}$, maps $\begin{pmatrix} T \\ \downarrow \\ X \end{pmatrix} \to \begin{pmatrix} E \\ \downarrow \\ X \end{pmatrix}$ in $\mathcal{E}/X$ correspond to maps $\bar{k}\colon T \to \tilde{Z}^Y$ such that $\tilde{g}^Y \bar{k} = \bar{\theta} h$. Taking exponential transposes, such maps $\bar{k}$ correspond to maps $k\colon T \times Y \to \tilde{Z}$ such that the following square commutes:

$$\begin{tikzcd}
T \times Y \arrow[r, "k"] \arrow[d, "h \times 1_Y"'] & \tilde{Z} \arrow[d, "\tilde{g}"] \\
X \times Y \arrow[r, "\theta"'] & \tilde{X}
\end{tikzcd}$$

This means that the two partial maps $T \times Y \rightharpoonup X$, classified by the left-bottom path and the top-right path, are equal.

Consider the two pullback squares (where the vertical maps are monos)

$$\begin{tikzcd}
T \times_X Y \arrow[r, "\pi_1"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[r, "f"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d] \\
T \times Y \arrow[r, "h \times 1_Y"'] & X \times Y \arrow[r, "\theta"'] & \tilde{X}
\end{tikzcd}$$

The partial map $T \times Y \rightharpoonup X$ classified by $\theta(h \times 1_Y)$ is given by the left and top edges of the rectangle.

On the other hand, the partial map $T \times Y \rightharpoonup X$ classified by $\tilde{g}k$ is given by the left and top edges of the rectangle

$$\begin{tikzcd}
U \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Z \arrow[r, "g"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d] \\
T \times Y \arrow[r, "k"'] & \tilde{Z} \arrow[r, "\tilde{g}"'] & \tilde{X}
\end{tikzcd}$$

where both squares are again pullbacks and the vertical maps are monos.

If both these partial maps are the same, we must have $U \cong T \times_X Y$. So maps k correspond to maps $k' : T \times_X Y \to Z$ such that

$$(T \times_X Y \xrightarrow{k'} Z \xrightarrow{g} X) = (T \times_X Y \xrightarrow{\pi_1} Y \xrightarrow{f} X)$$

That is, to maps $\begin{pmatrix} T\times_X Y \\ \downarrow \\ X \end{pmatrix} \to \begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix}$ in $\mathcal{E}/X$. ■

Exercise 3.4.4 What's the intuition behind this construction of exponentials? ■

Note that there is a natural bijection

$$\frac{X \to Y \in \mathcal{E}}{1 \to X^*Y \in \mathcal{E}/X}$$

That is 'elements' of Y (i.e. X^*Y) in $\mathcal{E}/X$ correspond to elements of Y in $\mathcal{E}$, 'varying over X'.

The above theorem has been called the fundamental theorem of topos theory. It underpins the whole notion of the internal language of a topos, and has many other consequences besides. In understanding it, $\mathcal{S}ets$ provides some useful motivation:

Example 3.4.5 Recalling that $\mathcal{S}ets/X \simeq \mathcal{S}ets^X$, we see that X^* corresponds to the diagonal functor $\mathcal{S}ets \to \mathcal{S}ets^X$ that sends a set Y to the constant family

$$\{Y\}_{x\in X}$$

And given $\begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix} \in \mathcal{S}ets/X$, which corresponds to a family $\{Z_x\}_{x\in X}$, we have

$$\Sigma_X \begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix} = \coprod_{x\in X} Z_x \qquad \Pi_X \begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix} = \prod_{x\in X} Z_x$$

This is because the operation of taking X-indexed products (or coproducts) is by definition the right (resp. left) adjoint to the diagonal functor. ■

So the functors Σ_I and Π_I can be regarded as computing I-indexed coproducts and products in $\mathcal{E}$. The theorem also has an easy, but important, generalisation.

Definition 3.4.6 *Let $\mathbb{B}$ be a category with pullbacks and let $f : Y \to X$ be a map in $\mathbb{B}$. Then we get a **pullback functor***

$$f^* : \mathbb{B}/X \to \mathbb{B}/Y$$

which takes an object $\begin{pmatrix} Z \\ \downarrow \\ X \end{pmatrix} \in \mathbb{B}/X$ to $\begin{pmatrix} Z\times_X Y \\ \downarrow \\ Y \end{pmatrix} \in \mathbb{B}/Y$, its pullback along f, and acts in the corresponding way on morphisms.

Corollary 3.4.7 *Let $\mathcal{E}$ be a topos, and let $f : Y \to X$ be a map in $\mathcal{E}$. Then*

- *the pullback functor $f^* : \mathcal{E}/X \to \mathcal{E}/Y$ is logical; and*
- *f^* has left and right adjoints $\Sigma_f, \Pi_f : \mathcal{E}/Y \to \mathcal{E}/X$.*

Proof. Observe that $\begin{pmatrix} Y \\ \downarrow f \\ X \end{pmatrix} \in \mathcal{E}/X$; it is simple to check that $(\mathcal{E}/X)/\begin{pmatrix} Y \\ \downarrow f \\ X \end{pmatrix} \simeq \mathcal{E}/Y$. So the corollary reduces to the previous theorem. ■

Exercise 3.4.8 If $\mathcal{E} = \mathcal{S}ets$, and identifying $\mathcal{S}ets/X$ with $\mathcal{S}ets^X$, describe Σ_f and Π_f explicitly. (See 2.3.6.) ■

The functors Σ_f, Π_f satisfy a certain compatibility condition called the **Beck-Chevalley condition**. We will rarely mention it explicitly, but it does play an important, if disguised, role in interpreting the internal language: see 2.3.5.

Exercise 3.4.9 Suppose we have a pullback square

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "h"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "k"] \\ Z \arrow[r, "g"'] & W \end{tikzcd}$$

Then there are canonical natural transformations

$$\Sigma_f h^* \Rightarrow k^* \Sigma_g, \qquad k^* \Pi_g \Rightarrow \Pi_f h^*$$

Cf. the discussion after definition 2.3.3. ■

Proposition 3.4.10 *These natural transformations are in fact natural isomorphisms.*

Proof. For the Σ-functors, this can be verified easily by direct calculation. The result for the Π-functors follows by playing around with adjunctions. ■

The Beck-Chevalley condition is just what we need to ensure that the formation of indexed products and coproducts—and, as we'll see in a minute, universal and existential quantification—commutes with weakening, and hence that their interpretation is independent of the extra free variables in the environment (i.e. does not change if we add 'dummy' free variables). We'll see it again soon.

Colimits are stable under pullback

Proposition 3.4.11 *If $f\colon X \to Y$ in a topos $\mathcal{E}$, f^* preserves colimits. In particular, in any pullback square*

$$\begin{tikzcd} Z \arrow[r, "k"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ W \arrow[r, "h"'] & Y \end{tikzcd}$$

if h is an epi then so is k.

Proof. f^* has a right adjoint Π_f, hence preserves any colimits that exist in $\mathcal{E}/Y$. For the second statement, note that $h: W \to Y$ is epi if and only if $Y \cong Y +_W Y$, where $Y +_W Y$ is the pushout of h along itself. But a square in $\mathcal{E}$ is a pushout if and only if applying the functor Y^* yields a pushout square in $\mathcal{E}/Y$ (similarly for X^*); and f^* preserves pushouts. ■

Here is a simple application.

Proposition 3.4.12 *If $X \in \mathcal{E}$ and there is a map $f: X \to 0$, then $X \cong 0$.*

Proof. $\left(\begin{smallmatrix} 0 \\ \downarrow 1_0 \\ 0 \end{smallmatrix}\right)$ is initial in $\mathcal{E}/0$, so $\left(\begin{smallmatrix} X \\ \downarrow 1_X \\ X \end{smallmatrix}\right) = f^*\left(\begin{smallmatrix} 0 \\ \downarrow 1_0 \\ 0 \end{smallmatrix}\right)$ is initial in $\mathcal{E}/X$; it follows that X is an initial object in $\mathcal{E}$. ■

Warning. It does *not* follow that

$$X^0 = \begin{cases} 1 & X \cong 0 \\ 0 & \text{otherwise} \end{cases}$$

For example, consider $X = \langle 0, 1 \rangle \in \mathcal{S}ets^2$.

Exercise 3.4.13 If $X, Y \in \mathcal{E}$, the coproduct injections

$$\mathsf{in}_0: X \to X + Y, \qquad \mathsf{in}_1: Y \to X + Y$$

are mono, and the square

$$\begin{tikzcd} 0 \arrow[r] \arrow[d] & X \arrow[d, "\mathsf{in}_0"] \\ Y \arrow[r, "\mathsf{in}_1"'] & X + Y \end{tikzcd}$$

is a pullback. (Hint: use exercise 3.3.8.) That is, coproducts in $\mathcal{E}$ are *disjoint.* ■

Image factorizations

Recall that if $f: X \to Y$ is function between sets, then f has an *image*, namely

$$\{y \in Y \mid f(x) = y \text{ for some } x \in X\} \subseteq Y$$

Kelly and Tierney showed that this generalises to an arbitrary topos:

Theorem 3.4.14 *Any map $f: X \to Y$ in a topos factors as an epi followed by a mono:*

$$X \overset{q}{\twoheadrightarrow} Q \overset{i}{\rightarrowtail} Y$$

*This is called the **image factorization** of f, and it is necessarily unique up to isomorphism. Informally, Q is often called the **image** of f.*

Proof. Let $a, b\colon R \rightrightarrows X$ be the kernel pair of f, and let $q\colon X \to Q$ be the coequalizer of this pair of maps. Intuitively $Q = X/\sim$ where

$$x \sim x' \quad \text{iff} \quad f(x) = f(x')$$

Since f coequalizes its kernel pair, we have a factorization

$$X \overset{q}{\twoheadrightarrow} Q \xrightarrow{i} Y$$

We need to show that i is mono. This is surprisingly hard.

Suppose we have maps $c, d\colon T \rightrightarrows Q$ such that $ic = id$; we have to show that $c = d$. Form the pullback square

$$\begin{tikzcd}
S \arrow[r, "e"] \arrow[d, "\langle g, h\rangle"'] \arrow[dr, phantom, very near start, "\lrcorner"] & T \arrow[d, "\langle c, d\rangle"] \\
X \times X \arrow[r, "q \times q"'] & Q \times Q
\end{tikzcd}$$

Now $(-) \times Q$ has a right adjoint $(-)^Q$, so it preserves colimits and hence epis; so, since q is epi, $q \times 1_Q$ is epi. Similarly $1_X \times q$ is epi, and so

$$q \times q = (q \times 1_Q)(1_X \times q)$$

is epi. And so, by 3.4.11, e is epi.

Now, observe that

$$fg = iqg = ice = ide = iqh = fh$$

So, by the definition of 'kernel pair', there is a unique map $k\colon S \to R$ such that $g = ak, h = bk$. Then

$$ce = qg = qak = qbk = qh = de$$

But since e is epi, $c = d$. ■

Exercise 3.4.15 The image of a map $f\colon X \to Y$ is the largest subobject of Y through which f factors: that is, if f factors through $Y' \rightarrowtail Y$, then so does i (uniquely). ■

Corollary 3.4.16 *Suppose $f\colon X \to Y$ is epi; then it's the coequalizer of its kernel pair.*

Proof. Look back at the proof above. If f is epi, then so is i; but i is mono, so by 3.3.1 it's an isomorphism. ■

Interpreting the propositional connectives

It's time to explain why toposes model constructive logic. Recall that we can identify subobjects of X with predicates on X. The structure of $\mathrm{Sub}(X)$ thus tells us about the 'propositional logic' of X. For example, if $X \in \mathcal{S}ets$ then $\mathrm{Sub}(X)$ is a Boolean algebra, so the logic of X is classical.

Theorem 3.4.17 *Let $\mathcal{E}$ be a topos. Then $Sub(1)$ is a Heyting algebra.*

Proof. (Tim Brook showed me this proof.) Recall that $\mathrm{Sub}(1)$ is equivalent to $\mathrm{Mono}(1)$, the full subcategory of $\mathcal{E}$ on the objects U such that $U \rightarrowtail 1$.

First note that $\mathrm{Mono}(1)$ is a reflective subcategory of $\mathcal{E}$. For given $X \in \mathcal{E}$ let its **support** $\sigma X \rightarrowtail 1$ be the image of the unique map $X \to 1$. By exercise 3.4.15, σ is functorial. It's easy to check that it's a reflection functor.

Now, since $\mathcal{E}$ has finite products and coproducts and $\mathrm{Mono}(1)$ is a full reflective subcategory, $\mathrm{Mono}(1)$ has them too; so $\mathrm{Sub}(1)$ has finite meets and joins. Note that this gives us a recipe for constructing them:

$$U_1 \cap U_2 = U_1 \times U_2, \qquad U_1 \cup U_2 = \sigma(U_1 + U_2)$$

Finally, if $X \in \mathcal{E}$, $(-)^X$ has a left adjoint $(-) \times X$, and hence preserves monos. So, since $1^X \cong 1$, if $U \rightarrowtail 1$ then $U^X \rightarrowtail 1$. In particular, $\mathrm{Mono}(1)$ is closed under exponentials in $\mathcal{E}$, and hence we can put

$$U_1 \to U_2 = {U_2}^{U_1}$$

So $\mathrm{Sub}(1)$ is a Heyting algebra. ■

Corollary 3.4.18 *Let $\mathcal{E}$ be a topos, $X \in \mathcal{E}$. Then $Sub(X)$ is a Heyting algebra.*

Proof. This is the previous theorem, applied to the topos $\mathcal{E}/X$. ■

Exercise 3.4.19 Show that if $X_1, X_2 \rightarrowtail X$ then $X_1 \cap X_2 \rightarrowtail X$ is constructed as the pullback

$$X_1 \times_X X_2$$

and $X_1 \cup X_2 \rightarrowtail X$ is the image of

$$X_1 + X_2 \to X$$

Can you give an explicit description of $X_1 \to X_2$? ■

We can say more. Recall that lattices, Boolean algebras etc. can be characterised equationally. The same holds for Heyting algebras, so the following definition makes sense.

Definition 3.4.20 *An **internal Heyting algebra** in a category with products is an object H equipped with maps*

$$\top : 1 \to H, \qquad \bot : 1 \to H$$

$$\wedge : H \times H \to H, \qquad \vee : H \times H \to H, \qquad \to : H \times H \to H$$

satisfying a number of axioms which we don't bother to state.

There are actually various ways to formulate the axioms. We have omitted them, since there are quite a few, and since we won't be using them in the sequel anyway. See, e.g. [12, 5.12].

Corollary 3.4.21 *In any topos, Ω is an internal Heyting algebra.*

Proof. We have just seen that for each $X \in \mathcal{E}$, $\mathrm{Hom}_{\mathcal{E}}(X, \Omega) \cong \mathrm{Sub}(X)$ is a Heyting algebra. The result follows by the Yoneda lemma. For example, since $\mathrm{Sub}(X) \cong \mathrm{Hom}_{\mathcal{E}}(X, \Omega)$ the natural transformation

$$\cap \colon \mathrm{Sub}(-) \times \mathrm{Sub}(-) \to \mathrm{Sub}(-)$$

(taking a pair of subobjects to its meet) gives us a natural transformation

$$\mathrm{Hom}_{\mathcal{E}}(-, \Omega \times \Omega) \cong \mathrm{Hom}_{\mathcal{E}}(-, \Omega) \times \mathrm{Hom}_{\mathcal{E}}(-, \Omega) \to \mathrm{Hom}_{\mathcal{E}}(-, \Omega)$$

and hence a map

$$\wedge \colon \Omega \times \Omega \to \Omega$$

■

Exercise 3.4.22 Construct $\wedge \colon \Omega \times \Omega \to \Omega$ directly, without using the Yoneda lemma. (Hint: what subobject of Ω does it classify?) ■

Exercise 3.4.23 What is an internal Boolean algebra? Is the subobject classifier in $\mathcal{S}ets^2$ an internal Boolean algebra? What about the subobject classifier in $\mathcal{S}ets^{\to}$? ■

Interpreting the quantifiers

Let $X \in \mathcal{E}$. Recall from the proof of 3.4.17 that we have an adjunction

$$\begin{tikzcd} \mathrm{Sub}(X) \arrow[r, shift right, "i_X"'] & \mathcal{E}/X \arrow[l, shift right, "\sigma_X"'] \end{tikzcd}$$

(We saw this in the case $X = 1$; the general result follows by passing to $\mathcal{E}/X$ as usual.)

Suppose $f \colon X \to Y$. We saw that the pullback functor

$$f^* \colon \mathcal{E}/Y \to \mathcal{E}/X$$

has left and right adjoints

$$\Sigma_f, \Pi_f \colon \mathcal{E}/X \to \mathcal{E}/Y$$

Definition 3.4.24 *Since f^* preserves monos, it restricts to a monotone function*

$$f^{-1} \colon Sub(Y) \to Sub(X)$$

This is called the ***inverse image map****.*

Proposition 3.4.25 *f^{-1} commutes with all the Heyting algebra operations.*

Proposition 3.4.26 *f^{-1} has left and right adjoints*

$$\exists_f, \forall_f \colon Sub(X) \to Sub(Y)$$

Proof. We identify subobjects of X with monos $m: X' \rightarrowtail X$:

$$\mathrm{Sub}(X) \simeq \mathrm{Mono}(X)$$

We let $\exists_f m = \sigma_Y \Sigma_f m$; i.e. it is the image of

$$X' \overset{m}{\rightarrowtail} X \xrightarrow{f} Y$$

And we let $\forall_f m = \Pi_f m$ (which is mono since Π_f has a left adjoint and hence preserves limits). It is routine to check that these do indeed define adjunctions. ∎

Exercise 3.4.27 Consider the so-called *doctrinal diagram* of Kock and Wraith:

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\mathcal{E}/X \arrow[r, shift left=4, "\Sigma_f"] \arrow[r, "f^* \;\perp" description] \arrow[r, shift right=4, "\Pi_f \;\perp" description] \arrow[d, shift right=2, "\sigma_X"'] \arrow[d, phantom, "\dashv"] & \mathcal{E}/Y \arrow[d, shift right=2, "\sigma_X"'] \arrow[d, phantom, "\dashv"] \\
\mathrm{Sub}(X) \arrow[u, shift right=2, "i_X"'] \arrow[r, shift left=4] \arrow[r, "\exists_f \;\perp" description] \arrow[r, shift right=4, "\forall_f"'] & \mathrm{Sub}(Y) \arrow[u, shift right=2, "i_X"']
\end{tikzcd}$$

(with $f^{-1}\;\perp$ labelling the arrow between $\exists_f$ and $\forall_f$)

Show that

$$\begin{array}{rcl}
f^* i_Y & \cong & i_X f^{-1} \\
\Pi_f i_X & \cong & i_Y \forall_f \\
\sigma_Y \Sigma_f & \cong & \exists_f \sigma_Y \\
\sigma_X f^* & \cong & f^{-1} \sigma_Y
\end{array}$$

Assuming f is mono, show that

$$\Sigma_f i_X \cong i_Y \exists_f$$

(None of this will be used in the sequel, at least not explicitly.) ∎

Example 3.4.28 If $f: X \to Y$ is a functions in $\mathcal{S}ets$, and $X' \subseteq X$

$$\begin{array}{rclcl}
\exists_f(X') & = & \{y \in Y \,|\, \exists x \in X.\, f(x) = y\} & = & f(X') \\
\forall_f(X') & = & \{y \in Y \,|\, \forall x \in X.\, f(x) = y \to x \in X'\} & &
\end{array}$$

This is thus a generalisation of example 3.1.12. ∎

So these left and right adjoints have quite a natural interpretation. In fact, we will only need to apply the proposition in the case when f is of the form

$$\pi_0: Y \times X \to Y$$

i.e. in a situation analogous to that described in example 3.1.12.

Let's see, then, how to interpret quantification. Suppose $X, Y \in \mathcal{E}$ and we have a predicate $\phi(y, x)$ on $Y \times X$:

$$\phi: Y \times X \to \Omega$$

Let $U \overset{m}{\rightarrowtail} Y \times X$ be the subobject of $Y \times X$ classified by ϕ. Applying $\exists_{\pi_0}, \forall_{\pi_0}$ to m gives us monos, which by abuse of notation we call

$$\exists_{\pi_0} U \rightarrowtail Y \qquad \forall_{\pi_0} U \rightarrowtail Y$$

Their classifying maps, which we call

$$\exists_{\pi_0}\phi, \forall_{\pi_0}\phi: Y \to \Omega$$

are the interpretations of $\exists x.\, \phi(y, x)$ and $\forall x.\, \phi(y, x)$.

Exercise 3.4.29 Use the Yoneda lemma to show that this definition of $\exists_{\pi_0}$ gives us a map

$$P(Y \times X) \to PY$$

which agrees with the one defined in exercise 3.3.10. ■

The Beck-Chevalley condition for quantifiers It's time to recall the funny condition from proposition 3.4.10. It's easy to prove an analogue for the functors $\exists_f, \forall_f$, but it's really only a special case that's of interest. Suppose $X, Y, Z \in \mathcal{E}$, so we have a pullback square

$$\begin{tikzcd}
Z \times Y \times X \arrow[r, "\pi_{12}"] \arrow[d, "\pi_{01}"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \times X \arrow[d, "\pi_0"] \\
Z \times Y \arrow[r, "\pi_1"'] & Y
\end{tikzcd}$$

Proposition 3.4.30 *If $\phi: Y \times X \to \Omega$ is a predicate on $Y \times X$, then*

$$\pi_1^* \exists_{\pi_0} \phi = \exists_{\pi_{01}} \pi_{12}^* \phi$$

and

$$\pi_1^* \forall_{\pi_0} \phi = \forall_{\pi_{01}} \pi_{12}^* \phi$$

The meaning of this is that "the interpretation of $\exists x$ and $\forall x$ commutes with weakening". To see this, let's work out what the left and right hand sides mean. $\exists_{\pi_0}$ takes the predicate

$$\phi(y, x)$$

on $Y \times X$ to the predicate

$$\exists x.\, \phi(y, x)$$

on Y; and $\pi_1^* \exists_{\pi_0} \phi: Z \times Y \to \Omega$ is the predicate

$$\exists x.\, \phi(y, x)$$

on $Z \times Y$. On the other hand, $\pi_{12}^*\phi: Z \times Y \times X \to \Omega$ is the predicate

$$\phi(y, x)$$

on $Z \times Y \times X$, and $\exists_{\pi_{01}} \pi_{12}^*\phi: Z \times Y \to \Omega$ should be the predicate

$$\exists x.\, \phi(y, x)$$

on $Z \times Y$ again. So it's natural that they're the same: in the syntax (in the internal language) they'll be indistinguishable.

3.5 Set-theoretic notation

Up till now we have been doing pure category theory. We gained some insight into the constructions by thinking of the objects of $\mathcal{E}$ as sets, but the proofs were all diagram chases. Sometimes this is a very succinct and attractive way of reasoning; but just as often, it's unintuitive and cumbersome. In this section, we'll see that there's an alternative way of proving things in a topos: namely, by using its *internal language*. This allows us to give a rigorous meaning in $\mathcal{E}$ to set-theoretic expressions and arguments.

The internal language is sometimes called the *Mitchell-Bénabou language*, after Mitchell and Bénabou. This early work was developed and refined during the 1970s by many people, such as Fourman, Joyal, Osius, Higgs and Rowbottom (to name only five); there are several versions around by now. The particular language presented here is novel, in that it includes dependent types and indexed products and coproducts in the style of Martin-Löf type theory. These are very useful in practice, and easy to add, but for some reason are usually left out.

The language we defined is not the most economical one possible. It's possible to make do with far fewer primitives, and to have a language that parallels the categorical axioms of a topos much more closely; this is the approach of the Bristol school, for example. However, I think that the system as it stands is about the simplest that would be useful for the purposes of the next chapter—and more generally—and all the primitives (except perhaps definite descriptions) have a direct categorical interpretation[5].

Rather than describe the language in full formal detail here, we will sketch a few aspects of it and give some examples of how it can be applied. If you don't like the notation, please feel free to change it[6].

[5]We should be careful about applying Occam's razor too overenthusiastically. Although, for example, $\exists$ can be defined in terms of $\forall$ and $\to$, to a category theorist this seems a little unnatural: the 'real' meaning of $\exists$ is given by the defining adjunction. Furthermore, there are other categorical settings—such as regular categories—in which $\forall$ and $\to$ cannot be interpreted, but where $\exists$ still makes perfectly good sense. However, I don't want to be dogmatic.

[6]We give a traditional 'category-theorist's view' of the internal language; a more 'logical' view is presented in Lambek/Scott and Bell, and it's very useful to compare the approaches. The logician's view of topos theory seems to be quite a valid one for all the computer science applications; but it is definitely not the case that "all of topos theory can be done in the internal language". Examples of situations where the internal language can't (yet) be used are: most results involving different toposes and geometric morphisms; 'large' categories relative to a topos; and certain special constructions within a single topos, such as the 'tiny' objects that crop up in the topos-theoretic account of differential forms. In such cases category theory, rather than set theory, is definitely the appropriate 'logic' to use—for the moment, at least.

Interpreting set-theoretic expressions and formulas

The internal language of a topos is a logical notation, together with a collection of rules, that let us talk about the objects and arrows in a topos as if we were dealing with a category of sets and functions. There are basically four kinds of things we can write down:

sets Expressions like $X \times Y$ or Y^X, where $X, Y \in \mathcal{E}$. These are meant to denote 'types' or 'sets' (objects of $\mathcal{E}$) and are interpreted by object in $\mathcal{E}$, in the obvious way.

terms Expressions like $f(x)$ (where $f: X \to Y \in \mathcal{E}$) or $\langle x, f(x)\rangle$. These are meant to denote 'elements' of sets. For example,

$$f(x) \in Y, \qquad \langle x, f(x)\rangle \in X \times Y$$

A term t of type Y, with free variables of types $X_1, \ldots, X_n$, is interpreted by a map

$$[\![t]\!]: X_1 \times \cdots \times X_n \to Y$$

One special case should be singled out: a term ϕ of type Ω, with a free variable x of type X (say) is called a *predicate* on X. Its interpretation is a map

$$[\![\phi]\!]: X \to \Omega$$

Clearly it classifies a subobject $\|\phi\| \rightarrowtail X$, the **extension** of ϕ.

equalities If t, t' are terms of type X, it makes sense to say

$$\mathcal{E} \models t = t'$$

This will be true if their interpretations in $\mathcal{E}$ are equal arrows.

logical deductions If ϕ is a predicate on X, it makes sense to say

$$\mathcal{E} \models \phi$$

This will be true if $[\![\phi]\!] = \top: X \to \Omega$, i.e. if $\|\phi\| \cong X$.

The way that types and terms are interpreted is generally rather obvious. For example, suppose we have a term t of type Y, and a term t' of type Z, and they both have a single free variable of type X. They denote maps

$$[\![t]\!]: X \to Y, \qquad [\![t']\!]: X \to Z$$

Then the term $\langle t, t'\rangle$ denotes the induced map

$$\langle [\![t]\!], [\![t']\!]\rangle: X \to Y \times Z$$

Substitution is modelled by composition. That is, suppose t is a term of type Y with a free variable x of type X, and s is a term of type X with a free variable z of type Z, so that $[\![t]\!]: X \to Y, [\![s]\!]: Z \to X$. Then $t[x := s]$ is a term of type Y with a free variable z of type Z, and

$$[\![t[x := s]]\!]: Z \to Y$$

is just $[\![t]\!][\![s]\!]$.

The interpretation of predicates (terms of type Ω) is a little more interesting. Suppose $\phi(x), \psi(x)$ are predicates on X, i.e.

$$[\![\phi]\!], [\![\psi]\!]: X \to \Omega$$

Then $\phi(x) \wedge \psi(x), \phi(x) \vee \psi(x), \phi(x) \to \psi(x)$ are interpreted using the Heyting algebra operations on

$$\mathrm{Sub}(X) \simeq \mathrm{Hom}_{\mathcal{E}}(X, \Omega)$$

or (equivalently) the internal Heyting algebra operations on Ω. For example, $[\![\phi \wedge \psi]\!]$ is the composite

$$X \xrightarrow{\langle [\![\phi]\!], [\![\psi]\!] \rangle} \Omega \times \Omega \xrightarrow{\wedge} \Omega$$

Similarly, suppose $\phi(y, x)$ is a predicate on $Y \times X$, so that

$$[\![\phi]\!]: Y \times X \to \Omega$$

then $\exists x : X.\, \phi(y, x), \forall x : X.\, \phi(y, x)$ are interpreted by

$$\exists_{\pi_0}[\![\phi]\!], \forall_{\pi_0}[\![\phi]\!]: Y \to \Omega$$

as described at the end of the previous section.

For every type X we have an equality predicate $\mathsf{Eq}(x, x')$ on $X \times X$; this is the classifying map $eq_X: X \times X \to \Omega$ of the diagonal subobject $\delta_X: X \rightarrowtail X \times X$.

Note that if $\phi(x)$ is a predicate on X, our language lets us write down a type

$$\{x : X \,|\, \phi(x)\}$$

This denotes the subobject

$$\|\phi\| \rightarrowtail X$$

classified by $[\![\phi]\!]: X \to \Omega$.

There are various rules that let us deduce things of the form

$$\mathcal{E} \models t = t'$$

or

$$\mathcal{E} \models \phi$$

In the second case, they are essentially the logical rules listed in section 3.1 (together with a couple of extra rules). That is, they are the rules of constructive logic. This is the sense in which toposes model 'constructive set theory'.

The above is just a sketch. It's slightly misleading on one point: it's actually better to interpret terms and formulas with free variables as arrows with domain 1, in slices of $\mathcal{E}$. That is, if t is a term of type Y with a free variable of type X, we interpret it by an arrow

$$[\![t]\!]: 1 \to [\![X]\!]^*[\![Y]\!]$$

in $\mathcal{E}/X$. This is, of course, equivalent to interpreting it as an arrow $[\![t]\!]: X \to Y$ in $\mathcal{E}$; the reason for using slices systematically to interpret variable environments is that it is technically convenient, especially when we are dealing with dependent types.

Full details of the language, and a fairly detailed outline of its interpretation, may be found in the appendix. Again, there are many different versions.

Example 3.5.1 Here is an informal argument in the internal language. Suppose we have $t : X$ and predicates ϕ_1, ϕ_2, ψ on X, and suppose that

$$\begin{array}{l} \mathcal{E} \models \phi_1(t) \\ \mathcal{E} \models \phi_2(t) \\ \mathcal{E} \models \phi_1(x) \wedge \phi_2(x) \to \psi(x) \end{array}$$

Then we can conclude that

$$\mathcal{E} \models \psi(t)$$

by applying the usual logical rules ($\wedge$-introduction and *modus ponens*). ∎

Example 3.5.2 Here is another informal argument in the internal language. Let $X \in \mathcal{E}$ and let $\phi(x)$ be any predicate on X. Let $\psi(y)$ be the unique predicate on 0 (i.e. ψ is the unique map $0 \to \Omega$). Then

$$\mathcal{E} \models \psi(y)$$

Furthermore,

$$\mathcal{E} \models \psi(y) \to \phi(x)$$

This is because $\psi(y) \to \phi(x)$ is a predicate on $X \times 0 \cong 0$, and hence is the unique predicate on 0. Therefore, applying *modus ponens*, we get

$$\mathcal{E} \models \phi(x)$$

Eh? Something's gone wrong here. ∎

What is wrong with this proof? The example shows that we have to be careful using the rule of *modus ponens*. The point is that $\phi(x)$ *is* always valid when thought of as a predicate on $X \times 0$; i.e. if we remember the variable y of type 0. If we forget about y, the reasoning is unsound. In the formal system described in the appendix, we deal with this difficulty by keeping careful track of free variables. When reasoning informally, and not worrying too much about free variables, the following formulation of *modus ponens*, which *is* sound, is useful.

Definition 3.5.3 *(Osius) The rule of* ***restricted modus ponens*** *is*

$$\textit{if } \mathcal{E} \models \phi \textit{ and } \mathcal{E} \models \phi \to \psi \textit{ then } \mathcal{E} \models \psi$$

provided every variable free in ϕ is also free in ψ.

Exercise 3.5.4 If you've read the appendix, show that the following 'commutations' for sums are derived rules.

$$\frac{[\Gamma]\ t : X + Y \quad [\Gamma, x : X]\ u : Z_0 \times Z_1 \quad [\Gamma, y : Y]\ u' : Z_0 \times Z_1}{[\Gamma]\ \pi_0 \mathsf{case}(t, x \triangleright u, y \triangleright u') = \mathsf{case}(t, x \triangleright \pi_0 u, y \triangleright \pi_0 u') : Z_0}$$

$$\frac{[\Gamma]\ t : X + Y \quad [\Gamma, x : X]\ u : Z_0 \times Z_1 \quad [\Gamma, y : Y]\ u' : Z_0 \times Z_1}{[\Gamma]\ \pi_1 \mathsf{case}(t, x \triangleright u, y \triangleright u') = \mathsf{case}(t, x \triangleright \pi_1 u, y \triangleright \pi_1 u') : Z_1}$$

$$\frac{[\Gamma]\ t : X + Y \quad [\Gamma, x : X]\ u : Z_0 + Z_1 \quad [\Gamma, y : Y]\ u' : Z_0 + Z_1 \quad [\Gamma, z_0 : Z_0]\ v : Z_2 \quad [\Gamma, z_1 : Z_1]\ v' : Z_2}{[\Gamma]\ \mathsf{case}(\mathsf{case}(t, x \triangleright u, y \triangleright u'), z_0 \triangleright v, z_1 \triangleright v') = \mathsf{case}(t, z_0 \triangleright \mathsf{case}(u, x \triangleright v, y \triangleright v'), z_1 \triangleright \mathsf{case}(u', x \triangleright v, y \triangleright v')) : Z_2}$$

$$\frac{[\Gamma]\ t : X + Y \quad [\Gamma, x : X]\ u : {Z_1}^{Z_0} \quad [\Gamma, y : Y]\ u' : {Z_1}^{Z_0} \quad [\Gamma]\ v : Z_0}{[\Gamma]\ \mathsf{case}(t, x \triangleright u, y \triangleright u')(v) = \mathsf{case}(t, x \triangleright u(v), y \triangleright u'(v)) : Z_1}$$

What do they mean? ∎

Spelling out the interpretation

To see why the internal language 'works', let's have a closer look at what's going on. Suppose $\phi(x), \psi(x)$ are predicates on X, whose extensions are

$$\|\phi\| \rightarrowtail X, \qquad \|\psi\| \rightarrowtail X$$

Then since $\wedge$ is interpreted using the meet in $\mathrm{Sub}(X)$, $[\![\phi(x) \wedge \psi(x)]\!]: X \to \Omega$ classifies

$$\|\phi\| \cap \|\psi\| \rightarrowtail X$$

Similarly, $[\![\phi(x) \vee \psi(x)]\!]: X \to \Omega$ classifies

$$\|\phi\| \cup \|\psi\| \rightarrowtail X$$

(i.e. the image of $\|\phi\| + \|\psi\| \to X$; see exercise 3.4.19.).

What's more, suppose that

$$\mathcal{E} \models \phi(x) \to \psi(x)$$

This means that $(\|\phi\| \to \|\psi\|) \rightarrowtail X$ is isomorphic to X; and this is clearly true if and only if

$$\|\phi\| \leq \|\psi\|$$

Exercise 3.5.5 If $\phi(y, x)$ is a predicate on $Y \times X$, show that

$$[\![\exists x : X.\, \phi(y, x)]\!]: Y \to \Omega$$

classifies the image of

$$\|\phi\| \rightarrowtail Y \times X \xrightarrow{\pi_0} Y$$

■

Our language has one very convenient feature: *definite descriptions.* Suppose $\phi(x)$ is a predicate on X, and

$$\mathcal{E} \models \exists x : X.\, \phi(x)$$

It does *not* follow that there is a map $x_0: 1 \to X$ such that $[\![\phi(x)]\!]x_0 = \top$. Intuitively, this is because the axiom of choice is not constructively valid.

Example 3.5.6 Let $\mathcal{E} = \mathcal{S}ets^{\to}$, and let $X = (2 \to 1), Y = (2 \xrightarrow{1_2} 2) \in \mathcal{E}$. There is an obvious map $e: Y \twoheadrightarrow X$, giving us an object $Z = \begin{pmatrix} Y \\ \downarrow e \\ X \end{pmatrix} \in \mathcal{E}/X$. Since e is epi, $\sigma Z = 1$ and so

$$\mathcal{E}/X \models \exists z : Z.\mathsf{true}$$

But if there were a map $1 \to Z \in \mathcal{E}/X$, we would have a splitting $f: X \to Y \in \mathcal{E}$ for e; and no such splitting exists. ■

However, if

$$\mathcal{E} \models \exists! x : X.\, \phi(x)$$

then Kock showed that such a map x_0 must exist. We denote it by

$$\mathsf{the}\ x\ \mathsf{such\ that}\ \phi(x)$$

'the unique $x \in X$ such that $\phi(x)$'. The existence of x_0 is not at all obvious. One way of constructing it is given in the appendix. There are others, though of course they all give the same answer.

Definite descriptions are often a very useful way of defining functions in the internal language. Namely, if $\phi(x, y)$ is a predicate on $X \times Y$ and we can show that

$$\mathcal{E} \models \forall x : X\, \exists! y : Y.\, \phi(x, y)$$

then there is a map $f\colon X \to Y$ such that

$$\mathcal{E} \models \forall x : X.\, \phi(x, f(x))$$

Namely, we let

$$f = \lambda x.\, \mathsf{the}\ y\ \mathsf{such\ that}\ \phi(x, y)$$

(More precisely, we have

$$[\![\mathsf{the}\ y\ \mathsf{such\ that}\ \phi(x, y)]\!]\colon 1 \to X^*Y \in \mathcal{E}/X$$

which gives us a map $X \to Y \in \mathcal{E}$.)

Exercise 3.5.7 It was Russell or Hilbert or some other old buffer who introduced definite descriptions into logic; the notation he used was

$$\iota x.\, \phi(x)$$

For arbitrary predicates ϕ on X, we also have Hilbert's **indefinite description** operator[7]:

$$\epsilon x.\, \phi(x)$$

This could also be written $\mathsf{any}\ x\ \mathsf{such\ that}\ \phi(x)$. It denotes "any x such that $\phi(x)$, if one exists", and is often quite convenient to reason with. If $\exists x.\, \phi(x)$ then one has

$$\phi(\epsilon x.\, \phi(x))$$

This looks quite outrageous and seems to give us, for example, the axiom of choice. But explain how, in the internal language of a topos, we can in fact reason about "an arbitrary $x' : X$ such that $\phi(x')$, if one exists" by adding a free variable

$$x' : \{x : X \mid \phi(x)\}$$

to the environment. This is one of the reasons why having subset types is so convenient. ■

Category theory, set theory and the internal language

The internal language can be used to prove category-theoretic facts about $\mathcal{E}$; the point being that a categorical property usually has the 'set-theoretic' characterisation you might expect it to have. Here are a couple of very simple examples, which are already quite useful. As mentioned in the appendix, for the sake of readability we will write "$x = x$" instead of $\mathsf{Eq}(x, x')$; we also write "$\|\phi\|$" instead of $[\![\{x : X \mid \phi(x)\}]\!]$.

[7] Introduced for reasons, related to finitism, which do not concern us here.

Proposition 3.5.8 *Let $f: X \to Y$ be an arrow in $\mathcal{E}$. Then f is mono iff*

$$\mathcal{E} \models \forall x, x' : X.\, f(x) = f(x') \to x = x'$$

Proof. By assumption,

$$[\![\forall x, x' : X.\, f(x) = f(x') \to x = x']\!] = \top : 1 \to \Omega$$

in $\mathcal{E}$. This implies that

$$[\![f(x) = f(x') \to x = x']\!] = \top : 1 \to \Omega$$

in $\mathcal{E}/X \times X$; and, by definition of Heyting implication, this says that

$$\|f(x) = f(x')\| \leq \|x = x'\|$$

regarded as subobjects of $X \times X$. In other words, since $\|x = x'\| \leq \|f(x) = f(x')\|$ always holds,

$$\|f(x) = f(x')\| = \|x = x'\| = X \xrightarrow{\delta_X} X \times X$$

But the classifying map of $\|f(x) = f(x')\|$ is $X \times X \xrightarrow{f \times f} Y \times Y \xrightarrow{eq_Y} \Omega$. So we have a pullback square

$$\begin{tikzcd}
\|f(x) = f(x')\| \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "\delta_Y"] \\
X \times X \arrow[r, "f \times f"'] & Y \times Y
\end{tikzcd}$$

Substituting X for $\|f(x) = f(x')\|$ in this square, and δ_X for the left-hand vertical arrow, we see after a quick diagram chase that f must be mono.

Furthermore, all the implications can be reversed. ■

As you can see, unwinding the definitions is tedious but fairly mechanical.

Proposition 3.5.9 *Let $f: X \to Y$ be an arrow in $\mathcal{E}$. Then f is epi iff*

$$\mathcal{E} \models \forall y : Y.\, \exists x : X.\, f(x) = y$$

Proof. By assumption,

$$[\![\exists x : X.\, f(x) = y]\!] = \top : 1 \to \Omega$$

in $\mathcal{E}/Y$.

Now by 3.5.5, $\|\exists x : X.\, f(x) = y\| \rightarrowtail Y$ is the image of

$$\|f(x) = y\| \rightarrowtail X \times Y \xrightarrow{\pi_1} Y$$

But $\|f(x) = y\| \rightarrowtail X \times Y$ is just $\langle 1_X, f\rangle : X \to X \times Y$, so our hypothesis tells us that the image of

$$X \xrightarrow{\langle 1_X, f\rangle} X \times Y \xrightarrow{\pi_1} Y$$

is Y. In other words, the image of $f: X \to Y$ is Y.

But the image of f is the coequalizer of its kernel pair; so f coequalizes its kernel pair, and hence is epi. Again, all the implications can be reversed. ■

Exercise 3.5.10 Let $f: X \to Y$ be an arrow in $\mathcal{E}$. Show directly (i.e. without using the previous two propositions) that if

$$\mathcal{E} \models \forall y : Y. \exists! x : X. f(x) = y$$

then f is an isomorphism. ■

Exercise 3.5.11 Suppose ϕ_0, ϕ_1 are predicates on X and

$$\mathcal{E} \models \phi_0 \leftrightarrow \phi_1$$

In the internal language, construct an explicit isomorphism $\|\phi_0\| \cong \|\phi_1\|$. ■

3.6 Arithmetic

So far we have not seen how to develop any mathematics—such as number theory—inside a topos. Indeed, the category of finite sets is a topos, so to be able to do any arithmetic we clearly need some extra assumptions. Actually, it turns out that we only need one additional axiom, which is very simple. It is due to Lawvere.

Definition 3.6.1 *A **natural numbers object** in a category $\mathbb{C}$ is an object N equipped with maps*

$$1 \xrightarrow{0} N \xrightarrow{s} N$$

with the following universal property: whenever we have an object X and maps

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

there is a unique map $g: N \to X$ such that the following diagram commutes:

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

It is clear that a natural numbers object must be unique up to isomorphism.

Example 3.6.2 In $\mathcal{S}ets$, the natural numbers object is $\mathbb{N}$, and $0, s$ are the zero and successor functions. Given $x \in X$ and $f: X \to X$, we get a function $g: N \to X$ taking

$$n \mapsto f^n(x)$$

We can see that the definition of natural numbers object captures the notion of *iteration*. ■

A remarkable amount of constructive mathematics—certainly most of elementary number theory—can be carried out in an arbitrary topos with a natural numbers object[8]. We won't do this, but it does seem worthwhile to show how to define some basic arithmetic functions, and to derive some basic properties of the natural numbers object.

Example 3.6.3 The addition map $+\colon N \times N \to N$ is defined as follows: consider the maps

$$1 \xrightarrow{\overline{1_N}} N^N \xrightarrow{s^N} N^N$$

These give rise to a map $\bar{+}\colon N \to N^N$, whose exponential transpose is $+$. ∎

This is just the usual definition of addition in terms of primitive recursion. Note, for example, that the fact that

$$\mathcal{E} \models \forall n \in N.\, 0 + n = n$$

corresponds to the fact that

$$\begin{tikzcd} & N \arrow[d, "\bar{+}"] \\ 1 \arrow[ur, "0"] \arrow[r, "\overline{1_N}"'] & N^N \end{tikzcd}$$

commutes.

Exercise 3.6.4 Show that addition is associative and commutative. ∎

Example 3.6.5 The maps

$$N \xrightarrow{N^{!_N}} N^N, \quad N^N \xrightarrow{1_{N^N} \times \overline{1_N}} N^N \times N^N \cong (N \times N)^N \xrightarrow{+^N} N^N$$

give rise to a map $N \to N^N$ whose exponential transpose is

$$\dot{-}\colon N \times N \to N$$

i.e. the subtraction map. ∎

Arithmetic in a topos was extensively studied by Tim Brook in his thesis, written in the days before we had the internal language. It is interesting that the familiar notions of order and recursion have such elegant categorical counterparts.

A natural numbers object *in a topos* has an interesting alternative characterisation, due to Freyd, which leads to a proof of the Peano postulates.

Exercise 3.6.6 Show that

$$1 \xrightarrow{0} N \xleftarrow{s} N$$

is a coproduct diagram in $\mathcal{E}$. (Hint: first find maps

$$1 \longrightarrow 1 + N \longrightarrow 1 + N$$

making $1 + N$ a natural numbers object; it's immediate that $N \cong 1 + N$.) ∎

[8]To do a serious amount of analysis, the axiom of *dependent choices* is also required.

Applying exercise 3.4.13, it follows that s is mono; i.e. that

$$\mathcal{E} \models \forall n, n' \in N.\, s(n) = s(n') \rightarrow n = n'$$

Exercise 3.6.7 $N \rightarrow 1$ is the coequalizer of $1_N, s\colon N \rightrightarrows N$. ■

Theorem 3.6.8 (Freyd) *Let $\mathcal{E}$ be a topos and suppose we have $N \in \mathcal{E}$ with arrows*

$$1 \xrightarrow{0} N \xrightarrow{s} N$$

such that

$$1 \xrightarrow{0} N \xleftarrow{s} N$$

is a coproduct diagram and

$$N \rightrightarrows N \longrightarrow 1$$

(as above) is a coequalizer diagram. Then N is a natural numbers object.

Corollary 3.6.9 *Logical functors preserve natural numbers objects.*

Proof. Because logical functors preserve colimits. ■

Natural numbers objects have many nice properties. For instance, even though the logic of a topos is constructive, the logic of the natural numbers object is quite classical:

Proposition 3.6.10 *If N is a natural numbers object in a topos $\mathcal{E}$,*

$$\mathcal{E} \models \forall n, n' \in N.\, (n = n') \vee (n \neq n')$$

That is, equality of natural numbers is decidable.

Proof. It suffices to show that

$$\mathcal{E} \models \forall n \in N.\, (n = 0) \vee (n \neq 0)$$

i.e. that $0\colon 1 \rightarrowtail N$ is a complemented subobject of N. But since

$$1 \xrightarrow{0} N \xleftarrow{s} N$$

is a coproduct diagram, $s\colon N \rightarrow N$ is mono (as noted above) and the union of $0\colon 1 \rightarrowtail N$ and $s\colon N \rightarrowtail N$ is the whole of N. But by exercise 3.4.13, their intersection is the empty subobject $0 \rightarrowtail N$. ■

It follows that the order relation on N is decidable too.

The Peano postulates

Theorem 3.6.11 *The Peano postulates are valid in any topos with a natural numbers object.*

We've seen one of them already. The only one which is a bit tricky is induction; the easiest way to see that induction is valid is to use a general induction principle for initial T-algebras (due to Lehmann, Smyth and Plotkin). We need a preliminary definition.

Definition 3.6.12 *Let $T: \mathbb{C} \to \mathbb{C}$ be a functor. A **T-algebra** is an object $X \in \mathbb{C}$ equipped with a map $\alpha: TX \to X$. A **T-algebra morphism** from $\alpha: TX \to X$ to $\beta: TY \to Y$ is a map $f: X \to Y$ such that the following square commutes:*

$$\begin{tikzcd} TX \arrow[r, "Tf"] \arrow[d, "\alpha"'] & TY \arrow[d, "\beta"] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

*T-algebras and morphisms form a category T-**Alg**. An **initial algebra** for T is an initial object in the category T-**Alg**.*

Lambek showed that if $\alpha: TX \to X$ is an initial algebra, α must be an isomorphism.

Proposition 3.6.13 *(See [14]) Suppose $\alpha: TX \to X$ is an initial algebra. Then any T-algebra morphism $m: X' \to X$ which is mono must be an isomorphism.*

Proposition 3.6.14 *Let N be a natural numbers object in a category $\mathbb{C}$. Then $[0, s]: 1 + N \to N$ is an initial algebra for the functor $TX = 1 + X$ (and hence is an isomorphism).*

The proofs of these two results are easy diagram chases; they are left as an exercise.

Corollary 3.6.15 *Let N be a natural numbers object in a topos $\mathcal{E}$ and let ϕ be a predicate on N such that*

$$\begin{array}{l} \mathcal{E} \models \phi(0) \\ \mathcal{E} \models \phi(n) \to \phi(s(n)) \end{array}$$

Then $\mathcal{E} \models \forall n : N.\, \phi(n)$.

Proof. We need to show that $\|\phi\| \cong N$. It suffices to find a T-algebra structure on $\|\phi\|$ such that the mono $m: \|\phi\| \rightarrowtail N$ is a T-algebra morphism, where T is the functor taking $X \mapsto 1 + X$.

But the hypotheses tell us that the maps

$$0: 1 \to N, \qquad sm: \|\phi\| \to N$$

factor through $\|\phi\|$, and this tells us that the T-algebra structure on N restricts to one on $\|\phi\|$; and to see that m is a T-algebra morphism, note that the relevant square commutes iff

$$1 + \|\phi\| \stackrel{[0, sm]}{\longrightarrow} N$$

factors through $\|\phi\|$, which also follows from the hypotheses. ■

3.7 Extending the language

The internal language of $\mathcal{E}$, as we've defined it, is already quite useful. But it would be useful to be able to talk about other kinds of constructions, such as quotients of equivalence relations. In this section we'll see how some of these extra constructions can be defined within the basic language.

There's nothing sacred about the suggestions made here; there are quite a few alternatives to the approaches I've chosen. Rather, I just want to show that, even though partial maps and quotients don't appear explicitly in the language, it's still safe to reason about them set-theoretically—because they can be defined in the language in a natural way. The same remark applies to a whole range of other phenomena.

Partial functions

Definition 3.7.1 *If $Y \in \mathcal{E}$, the* ***partial map classifier*** *of Y is*

$$\tilde{Y} \equiv \{Y_0 : PY \mid \forall y, y' : Y. (y \in Y_0 \wedge y' \in Y_0) \to y = y'\}$$

Note that $[\![\tilde{Y}]\!] = \tilde{Y}$, so to speak; that is, our definition of the partial map classifier in the internal language coincides with the earlier, categorical, definition.

We would like to add some new rules to our proof system to be able to talk about partial map classifiers and partial functions. The following seem like plausible candidates.

$$\frac{[\Gamma]\ Y \text{ set}}{[\Gamma]\ \tilde{Y} \text{ set}}$$

$$\frac{[\Gamma]\ Y \text{ set}}{[\Gamma]\ \mathsf{undef} : \tilde{Y}} \qquad \frac{[\Gamma, x' : \{x : X \mid \phi(x)\}]\ t : Y}{[\Gamma, x : X]\ t \,\mathsf{if}\, \phi : \tilde{Y}}$$

$$\frac{[\Gamma]\ Y \text{ set}}{[\Gamma, y : Y, y' : \tilde{Y}]\ \mathsf{converge}(y', y) : \Omega}$$

$$\frac{[\Gamma]\ Y \text{ set}}{[\Gamma, y : Y]\ \vdash \neg\mathsf{converge}(\mathsf{undef}, y)} \qquad \frac{[\Gamma, x : \{x : X \mid \phi(x)\}]\ t : Y}{[\Gamma, x : X]\ \phi(x) \vdash \mathsf{converge}(t \,\mathsf{if}\, \phi, t)}$$

$$\frac{[\Gamma]\ Y \text{ set}}{[\Gamma, y_1, y_2 : Y, y' : \tilde{Y}]\ \mathsf{converge}(y', y_1) \wedge \mathsf{converge}(y', y_2) \vdash \mathsf{Eq}(y_1, y_2)}$$

$$\frac{[\Gamma, x : \{x : X \mid \phi(x)\}]\ t : Y}{[\Gamma, x : X]\ \neg\phi(x) \vdash \mathsf{Eq}(t \,\mathsf{if}\, \phi, \mathsf{undef})}$$

Intuitively, $\tilde{Y}$ is the set of 'possibly undefined' elements of Y, and

$$t(x) \,\mathsf{if}\, \phi(x) = \text{“}t(x) \text{ (if } \phi(x) \text{ holds)”}$$

while $\mathsf{converge}(y', y)$ means "the possibly undefined element y' *is* defined, and is equal to $y \in Y$".

Note that with these rules, a partial map $f : X \rightharpoonup Y$ does indeed give rise to a function $\tilde{f} : \tilde{Y}^X$. For if the domain $X' \rightarrowtail X$ of f is classified by $\phi : X \to \Omega$, then we can let

$$\tilde{f} = \lambda x^X. t(x) \,\mathsf{if}\, \phi(x) : \tilde{Y}^X$$

And a function $g : \tilde{Y}^X$ gives rise to a partial map $X \rightharpoondown Y$. For if we let

$$\phi(x) = \exists y : Y.\,\mathsf{converge}(g(x), y)$$

then we have a function

$$\lambda x^{X'}.\,(\mathsf{the}\,y\,\mathsf{such\ that}\,\mathsf{converge}(g(x), y)) : Y^{X'}$$

where $X' = \{x : X \mid \phi(X)\}$.

Let's make the following definitions:

$$\begin{array}{rcl} \mathsf{undef} & \equiv & \varnothing : \tilde{Y} \\ t\,\mathsf{if}\,\phi(x) & \equiv & \{y \in Y \mid \phi(x)\,\mathsf{andalso}\,y = t\} : \tilde{Y} \\ \mathsf{converge}(y', y) & \equiv & y \in y' : \Omega \end{array}$$

Note that we can't, in general, use "$\wedge$" instead of "andalso" in the definition of $t\,\mathsf{if}\,\phi(x)$; in particular, this wouldn't work for $t = \mathsf{the}\,y^Y \mathsf{such\ that}\,\psi(x, y)$, where $\phi(x) \vdash \exists! y : Y.\,\psi(x, y)$. This is because t is only well-formed in the context $x : \{x' : X \mid \phi(x')\}$, not in the context $x : X$. In fact, this was the main reason for including the connective andalso in the language.

Exercise 3.7.2 If we make these definitions, the rules we listed earlier are derived rules in the original system. ∎

Let's see how to apply this in practice, by showing that partial functions $X \rightharpoondown Y$ can be defined by recursion. Note that this is a purely set-theoretic argument. It's convenient to write "$s\!\downarrow = t$" for $\mathsf{converge}(s, t)$, and "$s\!\downarrow$" for $\exists y : Y.\,\mathsf{converge}(s, y)$. (The following example is not the most general way of obtaining a recursively defined partial function; it's just a simple illustration of the rules.)

Proposition 3.7.3 *Suppose $X, Y \in \mathcal{E}$ and $g : \tilde{Y}^X \rightarrow \tilde{Y}^X$, and*

$$\mathcal{E} \models \forall f : \tilde{Y}^X.\,\forall x : X.\,f(x)\!\downarrow \rightarrow g(f)(x) = f(x)$$

Then

$$\mathcal{E} \models \exists f : \tilde{Y}^X.\,g(f) = f$$

Proof. The usual proof works. Namely, let

$$u = \lambda x.\mathsf{undef} : \tilde{Y}^X, \qquad g^n(u) = \mathsf{it}(g, u)(n)$$

where $n : \mathbb{N}$. Let

$$X' = \{x : X \mid \exists n : \mathbb{N}.\,g^n(u)(x)\!\downarrow\}$$

Then

$$\forall x : X'.\,\exists! y : Y.\,\exists n : \mathbb{N}.\,g^n(u)(x)\!\downarrow = y$$

Let

$$h = \lambda x^{X'}.\,\mathsf{the}\,y\,\mathsf{such\ that}\,\exists n : \mathbb{N}.\,g^n(u)(x)\!\downarrow = y$$

Then $g(\tilde{h}) = \tilde{h}$. ∎

Exercise 3.7.4 Formulate, and prove, that this defines the unique *least* fixed point of g. Deduce that there is a (least) fixed point operator $g \mapsto \mathsf{Y}(g)$. ■

Exercise 3.7.5 Show that every partial recursive function $\mathbb{N} \rightharpoonup \mathbb{N}$ can be represented as a term $t : \widetilde{\mathbb{N}}^{\mathbb{N}}$. But, show that there are many other terms, which do *not* correspond to partial recursive functions. That is, the notion of partial function we have defined is too flexible to capture the intuitive idea of 'computable partial function'; for it also admits partial functions computable relative to some 'definable' oracle. ■

Luckily, there *are* attractive ways of characterising the 'computable' partial functions in the internal language—even ways that generalise to sets other than $\mathbb{N}$. However, we won't discuss them here; this is, to an extent, still the subject of current research.

Example 3.7.6 If $f\colon X \rightharpoonup Y$ is a partial map and $t : X$, it's often useful to be able to reason about $f(t)$ as if it were an element of Y, without first checking that $f(t)$ is actually defined—though one would have to do this in the end. (This amounts to reasoning about $\epsilon y.\, f(t) = y$; cf. exercise 3.5.7.) Is this legitimate? Well, if one works in an environment

$$[\ldots, y_0 : \{y : Y \,|\, \mathsf{converge}(f(t), y)\}]$$

then one can think of y_0 as being "$f(t)$" and hence reason about $f(t)$ safely. ■

Quotients of equivalence relations

If $R \rightarrowtail X \times X$ is a relation, we write xRx' instead of $\langle x, x'\rangle \in R$.

Definition 3.7.7 *Suppose $X \in \mathcal{E}$ and $R \rightarrowtail X \times X$. Then R is an* ***equivalence relation*** *if*

- $\mathcal{E} \models \forall x : X.\, xRx$
- $\mathcal{E} \models \forall x, x' : X.\, xRx' \to x'Rx$
- $\mathcal{E} \models \forall x, x'x'' : X.\, xRx' \wedge x'Rx'' \to xRx''$

That is, if R is reflexive, symmetric and transitive.

Exercise 3.7.8 Formulate a categorical version of this definition. ■

From now on we will write "$\sim$" (instead of R) for an equivalence relation on an object X. Also, note that composing with the projection maps $\pi_0, \pi_1\colon X \times X \to X$ gives us a pair of maps $\sim \rightrightarrows X$. We would like to be able to talk about the quotient $X/\sim$ of X by the equivalence relation.

Exercise 3.7.9 If $f\colon X \to Y$, the kernel pair of f defines an equivalence relation on X. ■

Given a categorical equivalence relation, given as a pair of arrows, its 'quotient' is just their coequalizer. But one can also define the quotient of an equivalence relation set-theoretically.

Definition 3.7.10 *If $\sim$ is an equivalence relation on X, the* ***quotient of*** X ***by*** $\sim$ *is*

$$X/\sim\; \equiv \{X_0 : PX \,|\, \exists x : X.\, x \in X_0 \;\wedge\; \forall x, x' : X.\, x \in X_0 \to (x' \in X_0 \leftrightarrow x \sim x')\}$$

That is, $X/\sim$ is the set of $\sim$-equivalence classes. We will abuse notation by writing $X/\sim$ for the object $[\![X/\sim]\!] \in \mathcal{E}$.

Proposition 3.7.11 *$X/\sim$ really is the quotient of X by $\sim$.*

Proof. Let $\chi_\sim : X \times X \to \Omega$ classify $\sim \rightarrowtail X \times X$, and let $\overline{\chi_\sim} : X \to PX$ be its exponential transpose. Then

$$\overline{\chi_\sim}(x) = \{x' \in X \mid x \sim x'\}$$

Let $Q \rightarrowtail PX$ be the image of this map, so that we have a quotient map $q : X \twoheadrightarrow Q$. We can show that $Q \cong X/\sim$.

But we can deduce from [12, Proposition 1.23] that $\sim \rightrightarrows X$ is the kernel pair of $q : X \twoheadrightarrow Y$; and it follows from corollary 3.4.16 that $q : X \twoheadrightarrow Y$ is the coequalizer of $\sim \rightrightarrows X$. ■

What are suitable rules for reasoning about $X/\sim$? Here are some candidates:

$$\frac{[\Gamma, x, x' : X]\ x \sim x' : \Omega \quad [\Gamma] \vdash \text{“}{\sim} \text{ is an equivalence relation”}}{[\Gamma]\ X/\sim \ \text{set}}$$

$$\frac{[\Gamma, x, x' : X]\ x \sim x' : \Omega \quad [\Gamma] \vdash \text{“}{\sim} \text{ is an equivalence relation”}}{[\Gamma, x : X]\ \mathsf{q}_\sim x : X/\sim}$$

$$\frac{[\Gamma]\ f : Y^X \quad [\Gamma] \vdash \forall x, x' : X.\, x \sim x' \to \mathsf{Eq}(f(x), f(x'))}{[\Gamma]\ f_\sim : Y^{X/\sim}}$$

$$\frac{[\Gamma]\ f : Y^X \quad [\Gamma] \vdash \forall x, x' : X.\, x \sim x' \to \mathsf{Eq}(f(x), f(x'))}{[\Gamma]\ \lambda x.\, f_\sim(\mathsf{q}_\sim x) = f : Y^X}$$

$$\frac{[\Gamma]\ g : Y^{X/\sim} \quad [\Gamma]\ g' : Y^{X/\sim} \quad [\Gamma] \vdash \forall x, x' : X.\, x \sim x' \to \mathsf{Eq}(g(\mathsf{q}_\sim x), g'(\mathsf{q}_\sim x))}{[\Gamma]\ g = g' : Y^{X/\sim}}$$

We often write “$[x]_\sim$” instead of $\mathsf{q}_\sim x$, or just “$[x]$” if $\sim$ is understood.

These become valid if we make the following definitions:

$$\begin{array}{rcl} \mathsf{q}_\sim x & \equiv & \{x' \in X \mid x' \sim x\} \\ f_\sim & \equiv & \lambda z^{X/\sim}.\, (\textsf{the}\ y\ \textsf{such that}\ \forall x : X.\, x \in z \to f(x) = y) \end{array}$$

Exercise 3.7.12 What would we get if we left out the “$\exists$” clause in the definition of $X/\sim$? (This clause ensures that “every equivalence relation has a representative”, and so lets us reason about quotients using representatives.) ■

Exercise 3.7.13 Carry out a similar programme for *partial equivalence relations;* that is, symmetric, transitive (but not necessarily reflexive) relations on an object $X \in \mathcal{E}$. ■

We will be hearing a lot more about partial equivalence relations.

3.8 Category theory inside a topos

In the final section of the next chapter (section 4.6), we'll describe the topos-theoretic counterpart of the PER model: the category of *modest sets*. This is a full subcategory of $\mathcal{E}\!f\!f$ which is complete[9] (in an appropriate sense!) and hence provides us with the set-theoretic model for polymorphism that we were looking for all along. And it turns out to be nothing but the internal version of the PER model.

The material in that section relies on the theory of internal categories, internal full subcategories, notions of completeness etc. (Everything we will need is dealt with in [10], the main reference for what follows.) We will list the basic definitions and results; the explanations will mostly be a little terse. Future versions of these notes might conceivably contain a more comprehensive[10] treatment of all this material. However, if you're feeling lazy, note that *most of the results in section 4.6 can be understood without reading the present section*, at least in an informal way.

Internal families

Definition 3.8.1 *Let $I \in \mathcal{E}$. An **(internal) I-indexed family of objects of** $\mathcal{E}$ is an object $\begin{pmatrix} X \\ \downarrow f \\ I \end{pmatrix} \in \mathcal{E}/I$. We write*

$$X_i = f^{-1}\{i\} = \{x : X \mid f(x) = i\}$$

Note that, thought of as an expression in the internal language (with a free variable $i : I$) denoting a set, this indeed denotes $\begin{pmatrix} X \\ \downarrow f \\ I \end{pmatrix} \in \mathcal{E}/I$. We sometimes use the informal notation $(X_i)_{i:I}$ or just (X_i) for an internal family in a topos.

Given a fixed internal family (X_i), we would like to say what it means for another internal family (Y_j) to have the property that "each of the Y_j's is in the family (X_i)" (i.e. is isomorphic to one of the X_i's). There are two ways we might go about making this precise: using fibrations, or using the internal language. It would be good if the two definitions were equivalent.

Example 3.8.2 We could naively define a full subfibration of $\begin{matrix} \mathcal{E}^{\rightarrow} \\ \downarrow \\ \mathcal{E} \end{matrix}$ consisting of families $\begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix}$ such that there is a map $\phi: J \to I$ with $Y_j \cong X_{\phi(j)}$: more precisely, if there is a pullback square

$$\begin{tikzcd} Y \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\ J \arrow[r, "\phi"'] & I \end{tikzcd}$$

However, it turns out that the internal language does not allow us to assert the existence of such a map ϕ (though of course, for a *given* ϕ we can state the fact that the square is a pullback in

[9]The crucial proof is omitted; see [10] or my nonexistent notes.

[10]comprehensible?

the internal language). So this definition cannot be equivalent to one that can be formulated in the internal language. ■

In order to talk about "families whose objects are (isomorphic to) members of the family $(X_i)_{i:I}$" in the internal language, we need a slightly weaker notion. Namely, the closest we can get to asserting the existence of an isomorphism in the internal language is the following:

Definition 3.8.3 *If $X, Y \in \mathcal{E}$, we say that $\mathcal{E} \models$ "$X \cong Y$" if*

$$\mathcal{E} \models \exists f : Y^X.\ \text{"}f \text{ is an isomorphism"}$$

i.e.

$$\mathcal{E} \models \exists f : Y^X.\, \forall y : Y.\, \exists! x : X.\, f(x) = y$$

We sometimes say that X and Y are ***locally isomorphic****.*

Note that this does not quite imply that $X \cong Y$: since the axiom of choice may fail, the validity of this statement does not imply the (external) existence of an actual isomorphism $f\colon X \to Y$. In fact, there are counterexamples in sheaf toposes[11].

Definition 3.8.4 *Let $Y \in \mathcal{E}$. Then*

$$\mathcal{E} \models \text{"}Y \text{ is in the family } (X_i)_{i:I}\text{"}$$

means

$$\mathcal{E} \models \exists i : I.\, \text{"}Y \cong X_i\text{"}$$

Note that this definition also makes sense for a family $\begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix} \in \mathcal{E}/J$.

Exercise 3.8.5 Show that the class of families $\begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix}$ satisfying this condition forms a stable class of objects in the fibration $\begin{matrix} \mathcal{E}^{\to} \\ \downarrow \\ \mathcal{E} \end{matrix}$ (hence giving us a full subfibration). Show that this class is also definable. Is this true for the full subfibration defined previously? ■

Characterising the internally definable subfibration

In order to understand this definition better, we'd like to give it a fibrational characterisation independent of the internal language. We need to interpret some notions from the theory of fibrations in the special case of topos theory.

Lemma 3.8.6 $\begin{matrix} \mathcal{E}^{\to} \\ \downarrow \\ \mathcal{E} \end{matrix}$ *is locally small (cf. 2.4.14).*

Proof. Suppose $I \in \mathcal{E}$ and $(X_i), (Y_i) \in \mathcal{E}/I$. As in the case of $\mathcal{S}ets$, we let

$$[(X_i), (Y_i)] = \coprod_{i:I} {Y_i}^{X_i}$$

[11] Consider the two double covers of S^1 in $\mathrm{Sh}(S^1)$.

and let the generic map $u\colon (X_i) \to (Y_i) \in \mathcal{E}/[(X_i),(Y_i)]$ come from the interpretation of the term

$$[u : Z, x : X_{\pi(u)}]\ u(x) : Y_{\pi(u)}$$

Let's spell this out. Reverting to our earlier notation, we write

$$(X_i) = \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}, (Y_i) = \begin{pmatrix} Y \\ \downarrow \\ I \end{pmatrix}$$

and let $\begin{pmatrix} Z \\ \downarrow\pi \\ I \end{pmatrix} = \begin{pmatrix} Y \\ \downarrow \\ I \end{pmatrix}^{\begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}}$. Then $[(X_i),(Y_i)] = Z$. Furthermore, note that since $\pi^*\begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix} = \begin{pmatrix} X\times_I Z \\ \downarrow\pi_1 \\ Z \end{pmatrix}$, if we put

$$\begin{pmatrix} W \\ \downarrow k \\ I \end{pmatrix} = \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix} \times \begin{pmatrix} Z \\ \downarrow\pi \\ I \end{pmatrix} \in \mathcal{E}/I$$

then k factors through π; similarly for $\begin{pmatrix} Y \\ \downarrow \\ I \end{pmatrix}$.

So, consider the following map in $\mathcal{E}/I$:

$$\langle \mathrm{ev}, \pi_1\rangle\colon \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix} \times \begin{pmatrix} Z \\ \downarrow\pi \\ I \end{pmatrix} \longrightarrow \begin{pmatrix} Y \\ \downarrow \\ I \end{pmatrix} \times \begin{pmatrix} Z \\ \downarrow\pi \\ I \end{pmatrix}$$

This gives us the desired generic map $u\colon \pi^*\begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix} \to \pi^*\begin{pmatrix} Y \\ \downarrow \\ I \end{pmatrix} \in \mathcal{E}/Z$. ∎

Definition 3.8.7 *We say that* ***isomorphisms are representable*** *in a fibration* $\begin{smallmatrix} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{smallmatrix}$ *if for each* $I \in \mathbb{B}$ *and each* $X, Y \in \mathbb{E}_I$*, there is an object* $\mathrm{Iso}(X,Y) \in \mathbb{B}$*, a map* $\tau\colon \mathrm{Iso}(X,Y) \to \mathbb{B}$ *and an isomorphism*

$$\begin{tikzcd} & \tau^*X \arrow[r, "v"] \arrow[d, dotted, "p"] & \tau^*Y \\ I & \mathrm{Iso}(X,Y) \arrow[l, "\tau"] & \end{tikzcd}$$

which is a ***generic isomorphism*** $X \to Y$*: that is, given any* $\alpha\colon J \to I$*, and any isomorphism*

$$v\colon \alpha^*X \to \alpha^*Y \ \in \mathbb{E}_J$$

there is a unique map $\ulcorner f \urcorner\colon J \to \mathrm{Iso}(X,Y)$ *in* $\mathbb{B}$ *such that* $f = \ulcorner f \urcorner^* v$.

Note that if $\begin{smallmatrix} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{smallmatrix}$ is locally small, it is then the case that $\mathrm{Iso}(X,Y) \rightarrowtail [X,Y]$, and that the class of isomorphisms is definable. (This is really the only case which is of interest.)

Lemma 3.8.8 *Isomorphisms are representable in* $\begin{smallmatrix} \mathcal{E}^{\to} \\ \downarrow \\ \mathcal{E} \end{smallmatrix}$.

Proof. We simply let

$$\mathrm{Iso}((X_i),(Y_i)) = \{g : [(X_i),(Y_i)] \mid \text{“}g \text{ is an isomorphism”}\} \overset{m}{\rightarrowtail} [(X_i),(Y_i)]$$

$\tau = \pi m$, and $v = m^* u$. Note that since

$$\mathcal{E}/\mathrm{Iso}((X_i),(Y_i)) \models \text{“}v \text{ is an isomorphism”}$$

v is an isomorphism[12]. ■

Now we can give a fibrational characterisation of “belonging to a family (X_i)”. That is, the definition does correspond to a meaningful subfibration, and we can make a close connection between what goes on in the fibration $\begin{smallmatrix}\mathcal{E}^{\to}\\ \downarrow\\ \mathcal{E}\end{smallmatrix}$ and what goes on in the internal language of $\mathcal{E}$. This comes out when we study notions of equivalence and completeness, for example: [10].

Proposition 3.8.9 *Fix an internal family* $(X_i) = \begin{pmatrix} X\\ \downarrow\\ I\end{pmatrix}$ *and let* $(Y_j) = \begin{pmatrix} Y\\ \downarrow\\ J\end{pmatrix}$. *Then*

$$\mathcal{E}/J \models \text{“}Y_j \textit{ is in the family } (X_i)\text{”}$$

iff there is an epi $\phi\colon K \twoheadrightarrow J$, *a map* $\psi\colon K \to I$ *and a family* $\begin{pmatrix} Z\\ \downarrow\\ K\end{pmatrix}$ *such that we have two pullback squares*

$$\begin{tikzcd}
Y \arrow[d] & Z \arrow[l] \arrow[r] \arrow[d] \arrow[dl, phantom, "\llcorner", very near start] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d] \\
J & K \arrow[l, "\phi"'] \arrow[r, "\psi"'] & I
\end{tikzcd}$$

Proof. By abuse of notation we let $(Y_j), (X_i)$ denote the families

$$\pi_0^*\begin{pmatrix} Y\\ \downarrow\\ J\end{pmatrix}, \pi_1^*\begin{pmatrix} X\\ \downarrow\\ I\end{pmatrix} \in \mathcal{E}/(J \times I)$$

First note that, unwinding the interpretation of the internal language of a topos,

$$\mathcal{E}/(J \times I) \models \text{“}Y_j \cong X_i\text{”}$$

if and only if the map

$$\mathrm{Iso}((Y_j),(X_i)) \xrightarrow{\tau} J \times I$$

is epi. Therefore

$$\mathcal{E}/J \models \exists i : I.\, \text{“}Y_j \cong X_i\text{”}$$

[12] Note the connection between the notion of definability in the theory of fibrations, and the subobject classifier axiom in topos theory (which is what let us carve out $\mathrm{Iso}(X,Y) \rightarrowtail [X,Y]$ by defining a suitable predicate: both play the role of a comprehension axiom.

if and only if the map

$$\mathrm{Iso}((Y_j),(X_i)) \xrightarrow{\tau} J \times I \xrightarrow{\pi_0} J$$

is epi. Putting $K = \mathrm{Iso}((Y_j),(X_i))$ and

$$\begin{pmatrix} Z \\ \downarrow \\ K \end{pmatrix} = (\pi_0\tau)^* \begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix} \cong (\pi_1\tau)^* \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$$

we have the required two pullback squares.

Conversely, suppose we have $K, \phi, \psi, \begin{pmatrix} Z \\ \downarrow \\ K \end{pmatrix}$ as specified. Then obviously $\phi^* \begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix} \cong \psi^* \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$. So by the universal property of $\mathrm{Iso}((Y_j),(X_i))$, ϕ factors through $\mathrm{Iso}((Y_j),(X_i))$; and then, since ϕ is epi, $\pi_0\tau$ must be epi as well. ■

When (X_i) is the family of modest sets in $\mathcal{E}\!f\!f$ and J is an ω-set, this will turn out to be equivalent to the earlier, more naive fibrational definition of "belonging to the family (X_i)" given in example 3.8.2. This is not a trivial result and will not be proved in these notes.

There is a great deal more we could say here. Our results so far are actually special cases of a general theory of *stacks* and *descent* that goes back to Grothendieck[13], who invented it for completely different purposes. In the form most applicable to discussing $\mathcal{E}\!f\!f$ and modest sets, this theory is sketched in [10].

Generic subsets, relations, quotients

Here are some examples of internal families which we'll need later on. First, consider the membership relation (i.e. the evaluation map)

$$\in : X \times PX \to \Omega$$

Its exponential transpose $\bar{\in} : PX \to PX$ can be regarded as a PX-indexed family, i.e. an object of $\mathcal{E}/PX$. In fact, it is just the interpretation of

$$[\phi : \Omega^X]\ \{x \in X \mid \phi(x)\}$$

Let $X'_g = \begin{pmatrix} PX \\ \downarrow \bar{\in} \\ PX \end{pmatrix} \in \mathcal{E}/PX$. Then we have a mono

$$X'_g \rightarrowtail (PX)^* X \in \mathcal{E}/PX$$

and this is the **generic subset of** X in the following sense:

Proposition 3.8.10 *Consider the fibration* $\begin{matrix} \mathbb{E} \\ \downarrow \\ \mathcal{E} \end{matrix}$*, where*

$$\mathbb{E}_I = \textit{(monos } X' \rightarrowtail I^*X \textit{ in } \mathcal{E}/I\textit{, and commutative triangles)}$$

$X'_g \rightarrowtail PX^*X$ *is a generic object for this fibration.*

[13](I think)

This is just a restatement of the fact that a subobject of I^*X (i.e. a relation $X' \rightarrowtail I \times X$) is classified by a unique map $I \to PX$: see 3.2.18.

Similarly, let

$$E(X) = \{R : P(X \times X) | \text{ “}R \text{ is an equivalence relation”}\} \rightarrowtail P(X \times X)$$

Form the pullback

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & P(X \times X) \arrow[d, "\bar{\in}"] \\
E(X) \arrow[r] & P(X \times X)
\end{tikzcd}$$

Let $R_g = \left(\begin{smallmatrix} \bullet \\ \downarrow \\ E(X) \end{smallmatrix}\right)$. Then

$$R_g \rightarrowtail E(X)^*(X \times X) \in \mathcal{E}/E(X)$$

is the **generic equivalence relation on** X. Its quotient, the coequalizer

$$R_g \rightrightarrows E(X)^*X \twoheadrightarrow Q_g$$

is the **generic quotient of** X in the following sense:

Definition 3.8.11 *Let $X \in \mathcal{E}$. A **quotient** of X in $\mathcal{E}/I$ is an object $Q \in \mathcal{E}/I$ such that for some equivalence relation $R \rightarrowtail I^*X \times I^*X)$, $Q = (I^X)/R$: i.e. we have a coequalizer*

$$R \rightrightarrows I^*X \twoheadrightarrow Q$$

Proposition 3.8.12 *Q_g is a generic object for the full subfibration of $\begin{smallmatrix} \mathcal{E}^{\to} \\ \downarrow \\ \mathcal{E} \end{smallmatrix}$ on the quotients of X.*

Intuitively, Q_g is the internal family of quotients of X:

$$(X/R)_{R:E(X)}$$

As we saw earlier, this can be made precise in the internal language: it's the denotation of

$$[R : E(X)]\ X/R$$

Definition 3.8.13 *Let $Y \in \mathcal{E}$. Then we say that*

$$\mathcal{E} \models \text{“}Y \text{ is a quotient of } X \text{ by an equivalence relation”}$$

if

$$\mathcal{E} \models \text{“}Y \text{ is in the family } (X/R)\text{”}$$

Similarly for internal families $(Y_j) \in \mathcal{E}/J$.

The following characterization is almost immediate from the above remarks and proposition 3.8.9:

Proposition 3.8.14 *If* $(Y_j) = \begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix} \in \mathcal{E}/J$,

$$\mathcal{E} \models \text{“}Y \textit{ is a quotient of } X \textit{ by an equivalence relation”}$$

iff there is an epi $\phi\colon K \twoheadrightarrow J$ *and an equivalence relation* $R \rightarrowtail K^*X \times K^*X$ *in* $\mathcal{E}/K$ *such that*

$$\phi^* \begin{pmatrix} Y \\ \downarrow \\ J \end{pmatrix} = K^*X/R$$

Exercise 3.8.15 Are any of the generic objects mentioned above skeletal? ■

Of course, all this applies equally well to *partial* equivalence relations R on X. There is an alternative way to characterise quotients of partial equivalence relations on X which is often useful. We make a simple definition.

Definition 3.8.16 *Q is a **subquotient** of X iff there's a diagram*

$$X \leftarrowtail A \twoheadrightarrow Q$$

(The corresponding partial equivalence relation is given by the kernel pair of $A \twoheadrightarrow Q$.)

Then Q is a subquotient of X if and only if it's the quotient of X by a partial equivalence relation. The proof is left as a rather easy exercise.

Internal categories

Although the definition of internal category makes sense in any category with pullbacks, we will give a 'set-theoretic' definition using the internal language of $\mathcal{E}$. (However, our definition will make sense in the 'internal language' of a category with finite limits, suitably defined.)

Definition 3.8.17 *An **internal category** $\mathbf{C} \in \mathcal{E}$ consists of the following data:*

$$\begin{array}{ll} C_0, C_1 \in \mathcal{E} & \textit{(sets of objects and arrows)} \\ d_0, d_1\colon C_1 \rightrightarrows C_0 & \textit{(domain and codomain)} \\ id\colon C_0 \to C_1 & \textit{(identities)} \\ \circ\colon C_2 \to C_1 & \textit{(composition)} \end{array}$$

where

$$C_2 = \{\langle g, f\rangle : C_1 \times C_1 \mid d_1(f) = d_0(g)\}$$

This data must satisfy the usual axioms for a category:

$$\begin{array}{l} \mathcal{E} \models \forall c : C_0.\, d_0(id(c)) = d_1(id(c)) = c \\ \mathcal{E} \models \forall \langle f, g\rangle : C_2.\, (d_0(g \circ f) = d_0 g) \wedge (d_1(g \circ f) = d_1 f) \\ \mathcal{E} \models \forall f : C_1.\, id(d_0(f)) \circ f = f \circ id(d_1(f)) = f \\ \mathcal{E} \models \forall \langle h, g, f\rangle : C_3.\, (h \circ g) \circ f = h \circ (g \circ f) \end{array}$$

where

$$C_3 = \{\langle h, g, f\rangle : C_1 \times C_1 \times C_1 \mid d_1(f) = d_0(g), d_1(g) = d_0(h)\}$$

We often write $\mathbf{C} = (C_1 \rightrightarrows C_0)$ for an internal category, leaving the rest implicit.

Exercise 3.8.18 Show how to formulate the definition of internal category in any category with pullbacks (C_2 is constructed as a pullback and there is a commutative diagram for each of the axioms). ∎

Exercise 3.8.19 If $\mathbf{C}$ is an internal category in $\mathcal{E}$ and $I \in \mathcal{E}$, show how to define an internal category $I^*\mathbf{C} \in \mathcal{E}/I$ whose 'set' of objects is I^*C_0. ∎

Definition 3.8.20 *Let $\mathbf{C}, \mathbf{D} \in \mathcal{E}$ be internal categories. An **internal functor** $F: \mathbf{C} \to \mathbf{D}$ consists of the following data:*

$$F_0: C_0 \to D_0, \quad F_1: C_1 \to D_1$$

The usual axioms for a functor must be satisfied:

$$\begin{array}{l}
\mathcal{E} \models \forall f : C_1.\, F_0(d_0(f)) = d_0(F_1(f)) \wedge F_0(d_1(f)) = d_1(F_1(f)) \\
\mathcal{E} \models \forall c : C_0.\, id(F_0(c)) = F_1(id(c)) \\
\mathcal{E} \models \forall \langle g, f \rangle : C_2.\, F_1(g \circ f) = F_1(g) \circ F_1(f)
\end{array}$$

(where for simplicity, we have used the same symbols $d_0, d_1, id, \circ$ to denote the category operations in both $\mathbf{C}$ and $\mathbf{D}$).

Exercise 3.8.21 If $c \in C_0$, define the **constant functor** $\ulcorner c \urcorner: \mathbf{C} \to \mathbf{D}$. ∎

Exercise 3.8.22 What is an internal natural transformation? An internal adjunction? (Hint: use the characterisation of adjunctions in terms of having a unit and counit.) If F is an internal functor, write down a formula in the internal language that says "there exists a left adjoint for F". Note that this does *not* imply that F is part of an internal adjunction (i.e. that there is an internal functor that is internally a left adjoint for F). Explain why not. (You need not give a counterexample.) ∎

Exercise 3.8.23 Let $\mathbf{C}, \mathbf{D}$ be internal categories. If $F, G: \mathbf{C} \to \mathbf{D}$, what is $\mathrm{Nat}(F, G)$? Show how to define the internal functor category $[\mathbf{C}, \mathbf{D}]$. Hence show that the 2-category $\mathbf{cat}(\mathcal{E})$ of internal categories, functors and natural transformations in $\mathcal{E}$ is cartesian closed. ∎

Exercise 3.8.24 If $\mathbf{C} \in \mathcal{E}$ is an internal category and $I \in \mathcal{E}$, show how to define an internal category $I^*\mathbf{C} \in \mathcal{E}/I$. Do the same for internal functors and internal natural transformations. ∎

Externalisation

It is sometimes convenient to work with internal categories, and sometimes preferable to work with fibrations. Clearly it would be good to establish a connection between these two ways of talking about category theory 'relative to' $\mathcal{E}$, or more generally to a base category $\mathbb{B}$ with finite limits. The following definition is exactly analogous to the definition of $\mathrm{Fam}(\mathbb{C})$, for an ordinary small category $\mathbb{C}$: see definition 2.2.7 above[14].

[14]Except that the earlier definition made sense for all locally small categories and not just for small categories.

Definition 3.8.25 *Let* $\mathbf{C} = (C_1 \rightrightarrows C_0)$ *be an internal category in* $\mathcal{E}$*. The* ***externalisation*** *of* $\mathbf{C}$ *is the fibration*

$$\begin{array}{c} Fam_{\mathcal{E}}(\mathbf{C}) \\ \downarrow p \\ \mathcal{E} \end{array}$$

where $Fam_{\mathcal{E}}(\mathbf{C})$ *is the category with*

objects: *maps* $(a: I \to C_0)$*, where* $I \in \mathcal{E}$

arrows: *an arrow from* $(a: I \to C_0)$ *to* $(b: J \to C_0)$ *consists of a map* $\phi: I \to J$ *together with a map* $f: I \to C_1$ *such that*

$$\mathcal{E} \models \forall i : I.\, d_0(f(i)) = a(i)$$
$$\mathcal{E} \models \forall i : I.\, d_1(f(i)) = b(\phi(i))$$

and p *is the functor taking* $(a: I \to C_0)$ *to* I*.*

This notion of "internal I-indexed family of objects of $\mathbf{C}$" (i.e. a map $I \to C_0$) shouldn't be confused with our previous notion of "internal I-indexed family of objects of $\mathcal{E}$" (i.e. an object $\begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$ in the slice topos $\mathcal{E}/I$).

Exercise 3.8.26 Verify that $Fam_{\mathcal{E}}(C)$ is a category. Show that this construction yields a split fibration which is locally small and has a generic object. ■

Conversely, one can show that any locally small fibration $\begin{array}{c} \mathbb{E} \\ \downarrow p \\ \mathbb{B} \end{array}$ with a generic object is (equivalent to) the externalisation of an internal category in $\mathbb{B}$. Moreover, by 'enlarging the base category', *any* locally small fibration can be regarded as coming from an internal category: see [11, 1.4.8]. This may shed some further light on why local smallness is a natural condition.

Internal full subcategories The internal categories we'll be interested in are the *full subcategories* of $\mathcal{E}$ whose objects are the members of an internal family (X_i). It is not obvious that this makes sense. There is a purely categorical definition (see 2.38 of Johnstone's book on toposes) but again it seems more intuitive to use the internal language.

Suppose $(X_i) = \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$ is an internal I-indexed family of objects of $\mathcal{E}$. How could we define the (internal) "full subcategory" $\mathbf{C}$ of $\mathcal{E}$ on the objects in the family? Well, C_0 should be the set of (names of) objects in the family, i.e. $C_0 = I$; and given $i, j \in C_0$ (i.e. members X_i, X_j of the family), we should have $\mathrm{Hom}_C(i,j) = {X_j}^{X_i}$. This can be made precise as follows:

Definition 3.8.27 *Let* $(X_i) = \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$ *be an internal family in* $\mathcal{E}$*. The* ***internal full subcategory*** *on* (X_i) *is* $\mathbf{C} \in \mathcal{E}$*, where*

$$\begin{array}{rcl} C_0 & = & I \\ C_1 & = & \coprod_{k \in I \times I} {X_{\pi_1 k}}^{X_{\pi_0 k}} \quad = \quad \text{“}\coprod_{i,j \in I} {X_j}^{X_i}\text{”} \end{array}$$

and domain, codomain, identities and composition are defined in the obvious way, i.e.

$$\begin{array}{rcl} d_0(f) & = & \pi_0 \mathsf{which}(f) \\ d_1(f) & = & \pi_1 \mathsf{which}(f) \\ id(i) & = & \mathsf{in}_{\langle i,i \rangle} \lambda x^{X_i}.x \\ g \circ f & = & \mathsf{in}_{\langle d_0(f), d_1(g) \rangle} \lambda x^{X_{d_0(g)}}.g(f(x)) \end{array}$$

In the case $\mathcal{E} = \mathcal{S}ets$, this does indeed define the full subcategory of $\mathcal{S}ets$ whose family of objects is the family of sets $\{X_i\}_{i:I}$.

Exercise 3.8.28 Verify that this does indeed define an internal category. Show that its externalisation is equivalent to the full subfibration of $\begin{smallmatrix}\mathcal{E}^{\to}\\ \downarrow\\ \mathcal{E}\end{smallmatrix}$ described in example 3.8.2. (But note that the latter fibration is not split.) ■

Exercise 3.8.29 Show that the definition of internal full subcategory on an internal family $\left(\begin{smallmatrix}X\\ \downarrow\\ I\end{smallmatrix}\right)$ makes sense in any locally cartesian closed category, by expressing it in purely categorical terms; *or*, by expressing in purely in the language of Martin-Löf type theory, cf. [22]. ■

Notions of completeness

As pointed out in [10], there are different ways to define completeness for internal categories in a topos, and these are *not equivalent* as they are for ordinary categories. This is because the logic of a topos is constructive, not classical—in particular the axiom of choice fails. Cf. exercise 3.8.22. So we must we careful to distinguish between different notions of completeness[15].

We illustrate the distinctions using the simplest example possible.

Definition 3.8.30 *Let $\mathbf{C} \in \mathcal{E}$ be an internal category. We say that*

$$\mathcal{E} \models \text{“}\mathbf{C} \textit{ has a terminal object”}$$

or that $\mathbf{C}$ *has* ***a terminal object in the weak sense****, if*

$$\mathcal{E} \models \exists t : C_0.\ \text{“}t \textit{ is a terminal object in } \mathbf{C}\text{”}$$

where “t is a terminal object in $\mathbf{C}$” means

$$\forall c : C_0.\ \exists! f : C_1.\ d_0(f) = c \wedge d_1(f) = t$$

(i.e. for all c, there exists a unique arrow $f\colon c \to t$).

Definition 3.8.31 *We say that* $\mathbf{C}$ *has* ***a terminal object in the strong sense*** *if we have maps $t\colon 1 \to C_0$, $u\colon C_1 \to C_0$ such that $d_0 u = 1_{C_0}$ and the following square is a pullback:*

$$\begin{tikzcd} C_0 \arrow[r, "u"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C_1 \arrow[d, "d_1"] \\ 1 \arrow[r, "t"'] & C_0 \end{tikzcd}$$

(Intuitively, u “sends each $c \in C_0$ to the unique arrow $c \to t$”.)

[15] Note that as classical mathematicians, we are always free to use the axiom of choice in the *metatheory*, e.g. in asserting that we can always choose a cleavage for a fibration over $\mathcal{E}$. This has nothing to do with the logic of $\mathcal{E}$, as we are talking about ‘real’ fibrations, not internal ones.

This stronger definition says we have a *choice* of a terminal object of $\mathbf{C}$. This is not equivalent to the weaker definition because the axiom of choice may not hold. (If terminal objects were unique, we could use definite descriptions; but they're only unique up to isomorphism.) In fact, the stronger definition cannot be formulated in the internal language: cf. definition 3.8.3.

Exercise 3.8.32 Given $t: 1 \to C_0$ such that

$$\mathcal{E} \models \text{“}t \text{ is a terminal object in } \mathbf{C}\text{”}$$

show that there must be a suitable map $u: C_1 \to C_0$ as in the definition of terminal object in the strong sense. ∎

This distinction is the first subtlety we have to cope with. The second is the following: we may want to calculate, not just the limit of a *single* diagram in $\mathbf{C}$, but the limit of a *family* of diagrams in $\mathbf{C}$; furthermore, these diagrams may have different shapes. Just because we can compute the limits of single diagrams does not necessarily mean we can compute the limit of arbitrary families of diagrams.

To spell this out a bit, a diagram in $\mathbf{C}$ consists of an internal category $\mathbf{D} \in \mathcal{E}$ together with an internal functor

$$F: \mathbf{D} \to \mathbf{C} \in \mathcal{E}$$

On the other hand, if $I \in \mathcal{E}$, an I-indexed family of diagrams in $\mathbf{C}$ consists of an internal category $\mathbf{D} \in \mathcal{E}/I$ (i.e. an I-indexed family of internal categories, not necessarily all the same) together with an internal functor

$$F: \mathbf{D} \to I^*\mathbf{C} \in \mathcal{E}/I$$

(i.e. an I-indexed family of diagrams in $\mathbf{C}$). This is clearly a more general notion of 'diagram in $\mathbf{C}$'. We will have to take account of the difference in our definition.

Exercise 3.8.33 If $F \in [\mathbf{D}, \mathbf{C}]$ is an object of the internal functor category, show how to define the category $\mathrm{Cone}(F)$ of cones over F. (It is an internal subcategory of $[\mathbf{D}, \mathbf{C}]$.) ∎

Definition 3.8.34 *Let $I \in \mathcal{E}$ and let $\mathbf{D} \in \mathcal{E}/I$ be an internal category.* $\mathbf{C}$ ***has weak $\mathbf{D}$-limits*** *if*

$$\mathcal{E} \models \forall F : [\mathbf{D}, I^*\mathbf{C}].\ \text{“}\mathrm{Cone}(F) \textit{ has a terminal object”}$$

$\mathbf{C}$ ***has strong $\mathbf{D}$-limits*** *iff the diagonal functor*

$$\Delta: I^*\mathbf{C} \to [\mathbf{D}, I^*\mathbf{C}] \qquad c \mapsto \ulcorner c \urcorner$$

has a right adjoint $\varprojlim: [\mathbf{D}, \mathbf{C}] \to \mathbf{C}$.

That is, the *weak* definition asserts the validity of an existential assertion in the internal language, while the *strong* definition asserts the existence of an actual arrow in the topos $\mathcal{E}$.

Definition 3.8.35 *Let $I \in \mathcal{E}$.* $\mathbf{C}$ *is **weakly** (resp. **strongly**) **complete over** I if, for any $\mathbf{D} \in \mathcal{E}/I$, $I^*\mathbf{C}$ has weak (resp. strong) $\mathbf{D}$-limits.*

Notice the subtlety here. There is no way to speak, in the internal language, about diagrams indexed by an 'arbitrary' internal category: this is because, in the internal language, you can't quantify over all the objects in the topos, or all the internal categories. ("Quantification over sets, but not classes.") So in the definition of weak completeness, the quantifier "for all $\mathbf{D}$" is in the *metalanguage*. This means that weak completeness over I is not stable under pullback: if $\mathbf{C}$ is weakly complete over 1, $I^*\mathbf{C}$ need not be weakly complete over I—the logical functors I^* only preserve the validity of statements expressible in the internal language.

Definition 3.8.36 $\mathbf{C}$ *is* ***weakly*** *(resp.* ***strongly****)* ***complete*** *if, for any $I \in \mathcal{E}$, $\mathbf{C}$ is weakly (resp. strongly) complete over I.*

This is a lot of definitions, but it was not pointless to split them up. For example, it will turn out that the category of modest sets is weakly, but not strongly complete; however, it *is* strongly complete over certain objects I, namely the ω-sets.

Exercise 3.8.37 Define what it means for an internal category to have weak equalizers and (arbitrary) weak products, and show that an internal category is weakly complete iff it has weak equalizers and weak products. What about strong completeness? ■

It is the strong notions of completeness which tend to be more useful: for example, they are what is needed to model polymorphic type theories, or to construct particular left adjoints to functors. However, the weak notions are all we can talk about in the internal language (that is, if we are asserting the *existence* of limits, rather that working with properties of a *given* limit). So when we come to discuss the internal category of modest sets, we'll have to establish the connections between its strong and weak completeness properties.

Notions of completeness for fibrations 'Weak' and 'strong' completeness have analogues for fibrations. The fact that there is such an analogy, and the form it takes, should be compared with proposition 3.8.9: the idea is basically the same. We always assume that the base category $\mathbb{B}$ has pullbacks.

'Strong' completeness is just completeness in the sense of definition 2.3.5:

Definition 3.8.38 *A fibration* $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ *is* ***strongly complete*** *if it's complete in the usual sense.*

For internal categories, 'weak' corresponded to the validity of an existential assertion in the internal language. For fibrations, 'weak' truth means "true once you reindex along an epi". To see why these are analogous, cf. the proof of 3.8.8[16].

Definition 3.8.39 *A fibration* $\begin{array}{c}\mathbb{E}\\ \downarrow p\\ \mathbb{B}\end{array}$ *is* ***weakly complete*** *if*

- *given any finite diagram D in $\mathbb{E}_I$, there is an epi $\alpha\colon K \twoheadrightarrow I$ such that α^*D has a limit in $\mathbb{E}_K$, and reindexing functors preserve finite limits; and*
- *given any map $\phi\colon J \to I$, there is an epi $\alpha\colon K \twoheadrightarrow I$ such that if ψ is the map*

  $$\psi\colon J \times_I K \longrightarrow K$$

  obtained by pulling back ϕ along α, then ψ^* has a right adjoint Π_ψ, and these right adjoints satisfy the Beck-Chevalley condition (in the appropriate sense).

[16] Also, readers familiar with Kripke-Joyal semantics will appreciate this point already.

There are also strong and weak notions of equivalence, and not all notions of completeness are stable under all notions of equivalence, though you can probably guess which combinations work: see [10]. One can also show that for *definable* subfibrations of $\begin{array}{c}\mathcal{E}^{\to}\\ \downarrow\\ \mathcal{E}\end{array}$ the strong and weak notions coincide.

Finally, we have the following result that connects these notions of completeness for fibrations and for internal categories; it is stated but not proved in [10]. The proof is not short, but it is straightforward.

Proposition 3.8.40 *An internal category* $\mathbf{C} \in \mathcal{E}$ *is weakly complete if and only if the fibration* $Fam_{\mathcal{E}}(\mathbf{C})$ *is weakly complete.*

An analogous result holds for strong completeness—though unless we are prepared to allow base categories (such as ω-$\mathcal{S}ets$) which aren't toposes, no nontrivial examples are known. There are also similar results relating notions of equivalence between internal categories and between their externalisations.

Chapter 4

The effective topos

4.1 Recursive realizability

Classically, the logical connectives denote truth-functions, and we can explain their 'meaning' in terms of truth-tables. Although we have a similar semantics for constructive logic, using Heyting algebras, this is not the way constructive logic was originally meant to be understood.

The formalisation of constructive logic seems to begin with Glivenko, Kolmogorov (who invented the double negation translation) and Heyting in the 1920s[1]. Heyting put forward the following informal explanation of the connectives, which looks similar to one proposed by Kolmogorov. The idea is that to explain what a formula ϕ means, you have to say what it means to *prove* ϕ. It goes roughly as follows[2]:

A proof of...	...consists of...
$\phi \wedge \psi$	a proof of ϕ together with a proof of ψ
$\phi \vee \psi$	either a proof of ϕ or a proof of ψ, together with information telling us which of the two has been proved
$\phi \rightarrow \psi$	a procedure which, applied to any proof of ϕ, produces a proof of ψ
$\exists x.\, \phi(x)$	an object t, together with a proof of $\phi(t)$
$\forall x.\, \phi(x)$	a procedure which, given any object t, produces a proof of $\phi(t)$
$\perp$	nothing—there is no proof of $\perp$

This interpretation seems to be a good way of motivating recursive realizability[3]. Before we define what that is, we'll recall some definitions from elementary recursion theory. We'll also write down a version of first-order arithmetic is; this will give us a system to interpret that is simple without being trivial. It is only meant to serve as an example, and won't be needed after this section.

[1]Actually, I'm not certain about the history. There is an entertaining account in [1, Historical Appendix].

[2]Heyting's account was slightly more sophisticated. In particular, in the clauses for $\phi \wedge \psi$ and $\forall x.\, \phi(x)$, he also required a proof that the given procedure worked. But, that won't concern us.

[3]Though Kleene claims that it was *not* his original motivation. Interesting.

A version of constructive arithmetic

We will soup up constructive first-order logic (see the previous chapter) to produce a system of constructive arithmetic. The **terms** are given by

$$t \quad ::= \quad x \mid 0 \mid \mathsf{s}(t) \mid t + t' \mid t \times t'$$

and we add an atomic predicate symbol $t = t'$.

Equality rules These are the usual ones. (We have left the variable contexts out of all the rules, because they do not really play a role.)

$$\frac{}{\vdash x = x} \qquad \frac{}{u = u' \vdash t[x := u] = t[x := u']} \qquad \frac{}{\phi(t), t = t' \vdash \phi(t')}$$

Rules for arithmetic These are basically the Peano postulates plus some rules defining $+$ and $\times$. There are other ways to write them, of course.

$$\frac{}{\vdash \neg(\mathsf{s}(x) = 0)} \qquad \frac{}{\mathsf{s}(x) = \mathsf{s}(x') \vdash x = x'}$$

$$\frac{}{\vdash x + 0 = x} \qquad \frac{}{\vdash x + \mathsf{s}(x') = \mathsf{s}(x + x')}$$

$$\frac{}{\vdash x \times 0 = 0} \qquad \frac{}{\vdash x \times \mathsf{s}(x') = x + (x \times x')}$$

$$\frac{\vdash \phi(0) \quad \phi(x) \vdash \phi(\mathsf{s}(x))}{\vdash \phi(x)}$$

Exercise 4.1.1 Use induction to show that equality is decidable: $\vdash \forall x, y.\, x = y \vee \neg(x = y)$ ■

The realizability interpretation

Interesting mathematical statements tend to have some computational content. For example, the assertion that there are an infinite number of primes can be written

$$\forall x.\, \exists y.\, \text{“}y \text{ is prime”} \wedge y > x$$

and constructively speaking, this states that given any $x \in \mathbb{N}$, we can find a prime number $y > x$. The *algorithm* that produces y from x can be thought of as producing *evidence* justifying the statement.

This suggests that we can explain what the logical connectives mean, not by saying what it means to prove a formula ϕ, but by saying what it means for an algorithm to produce evidence justifying ϕ. Replacing ‘algorithm’ with ‘recursive function’ and tightening this up a little, we get the definition of recursive realizability. First we gather up some definitions from elementary recursion theory.

Definition 4.1.2 *If $n_0, n_1 \in \mathbb{N}$ then we define the coded pair*

$$\langle n_0, n_1 \rangle = (n_0 + 1)(n_0 + n_1 + 1)$$

There are corresponding primitive recursive projection functions

$$n \mapsto n_0 \qquad n \mapsto n_1$$

And if $n_1, \ldots, n_l \in \mathbb{N}$, we define the coded tuple

$$\langle n_1, \ldots, n_l \rangle \in \mathbb{N}$$

accordingly.

Definition 4.1.3 *A **primitive recursive predicate** is a predicate of the form*

$$f(x_1, \ldots, x_n) = 0$$

where f is primitive recursive. If $\phi(x_1, \ldots, x_n, x)$ is a primitive recursive predicate, let

$$\mu k.\, \phi(x_1, \ldots, x_n, k)$$

*denote the least k for which $\phi(x_1, \ldots, x_n, k)$ holds; if $\forall y.\, \neg\phi(x_1, \ldots, x_n, y)$, then $\mu k.\, \phi(x_1, \ldots, x_n, k)$ is undefined. This gives us a general recursive function of $x_1, \ldots, x_n$; it is said to be defined by **minimization**.*

Primitive recursive predicates can all be defined in our formal system and shown to be decidable (using the fact that equality is decidable). In fact, it is probably more convenient just to add a new function symbol for each primitive recursive function, together with suitable axioms; but as we won't be using the system, we won't bother.

Definition 4.1.4 *If $e, n \in \mathbb{N}$ then $\{e\}(n)$ denotes the result (if any) of applying the* eth *partial recursive function to n. And if $n_1, \ldots, n_l \in \mathbb{N}$, then*

$$\{e\}(n_1, \ldots, n_l) = \{e\}(\langle n_1, \ldots, n_l \rangle)$$

(when $l = 0$, by convention we let $\{e\}() = e$.) Also, if $t(x)$ is a computable expression containing the free variable x then $\Lambda x.t$ is an code for the partial recursive function $\lambda x.t$ (note that this definition implicitly relies on the S-m-n theorem).

Note that if we use $\{e\}(n)$ in an expression, we are implicitly imposing the condition that it be defined. For example, if we say something like

"suppose $e \in \mathbb{N}$ is a code such that for all $n \in \mathbb{N}$, $\{e\}(n)$ is even"

we mean, "suppose $e \in \mathbb{N}$ is a code such that $\{e\}(n)$ is defined for all $n \in \mathbb{N}$, and..."

The intuition behind recursive realizability is that a number $n \in \mathbb{N}$ 'realizes' a formula ϕ if n is a code for a recursive function that "produces evidence for the truth of ϕ". The following definition is not too precise. In particular, we have assumed that all closed terms evaluate to numerals and we are confusing numerals (in the theory) with actual numbers (outside, in its semantics)[4]. My excuse is that all this is just an informal way to motivate the construction of the effective topos and understand the way its internal logic works. It should not be taken too seriously.

[4]In fact, what follows is *not* what logician usually mean by 'realizability', namely a certain interpretation of an extension of arithmetic into itself, designed to prove certain metatheorems. But we are only interested in the semantic aspect of realizability, not in its application to the metamathematics of constructive systems.

Definition 4.1.5 *Let $[\vec{x}] = [x_1, \ldots, x_m]$ be an environment (i.e. a list of free variables), and let t be a term of first-order arithmetic all of whose free variables occur in the list $\vec{x}$; if $\vec{k} = k_1, \ldots, k_m \in \mathbb{N}^m$, let $t[\vec{x} := \vec{k}]$ denote the closed term gotten by substituting k_i for x_i ($i = 1, \ldots, m$). Similarly for formulas ϕ. Then we say that*

$n \in \mathbb{N}$ ***realizes...*** *(in environment $[\vec{x}]$)*	*...exactly when...*
$t = t'$	*for all $\vec{k} \in \mathbb{N}^m$, $t[\vec{x} = \vec{k}] = t'[\vec{x} = \vec{k}] = \{n\}(\vec{k})$*
$\phi \wedge \psi$	*$n = \langle n_0, n_1 \rangle$ where n_0 realizes ϕ and n_1 realizes ψ*
$\phi \vee \psi$	*$n = \langle 0, m \rangle$ where m realizes ϕ, or $n = \langle 1, m \rangle$ where m realizes ψ*
$\phi \to \psi$	*whenever m realizes ϕ, $\{n\}(m)$ realizes ψ*
$\exists x.\, \phi(x)$	*$n = \langle k, m \rangle$ where m realizes $\phi(k)$*
$\forall x.\, \phi(x)$	*for all k, $\{n\}(k)$ realizes $\phi(x)$*
$\perp$	*nothing—there is no realizer for $\perp$*

Recall that in the case $m = 0$ (i.e. in the empty environment) we say that $\{n\}() = n$ by convention; this means that if t, t' are *closed* terms, n realizes $t = t'$ in the empty environment iff $t = t' = n$.

Example 4.1.6 A realizer for

$$\forall x.\, \exists y.\, x + x = y$$

(in the empty environment) is a code $n \in \mathbb{N}$ such that for all $k \in \mathbb{N}$, $\{n\}(k) = \langle k', m \rangle$ where $k + k = k' = m$. For example we could take

$$n = \Lambda k.\, \langle 2k, 2k \rangle$$

(Note the 'irrelevant' extra copy of $2k$ that's needed.) ■

More generally, if $\phi = \forall x.\, \exists y.\, \psi(x, y)$ is a closed Π^0_2 formula—which we think of as a specification for a total function—then realizers for ϕ correspond to total recursive functions meeting that specification.

Exercise 4.1.7 Formalise the statement that there are an infinite number of primes and write out what it means for a number $n \in \mathbb{N}$ to realize it. Notice that, given x, the algorithm encoded by n produces not just y but a lot of other junk besides. ■

Definition 4.1.8 *An arithmetic formula ϕ is* ***realizable*** *if there is a number $n \in \mathbb{N}$ that realizes ϕ. More generally, a sequent*

$$\phi_1, \ldots, \phi_l \vdash \psi$$

is ***realizable*** *if we have $e \in \mathbb{N}$ such that whenever m_i realizes ϕ_i ($i = 1, \ldots, l$) then $\{e\}(m_1, \ldots, m_l)$ realizes ψ.*

Exercise 4.1.9 If $\vdash \neg(\phi \wedge \psi)$ then ϕ and ψ are not both realizable. ■

The realizability interpretation is a sound interpretation of constructive first-order arithmetic. The idea is that a proof of a sequent gives a recipe for constructing a realizer for it.

Proposition 4.1.10 *If $\vdash \phi$ is derivable then ϕ is realizable.*

Proof. This is basically straighforward: the idea is that the proof of $\vdash \phi$ gives us a recipe for constructing a realizer for ϕ. We will do a few clauses and leave the rest to the reader.

Suppose e realizes

$$\phi_1, \ldots, \phi_l \vdash \psi$$

and e' realizes

$$\phi_1, \ldots, \phi_l \vdash \psi'$$

Then $\Lambda m_1, \ldots, m_l.\langle\{e\}(m_1, \ldots, m_l), \{e'\}(m_1, \ldots, m_l)\rangle$ realizes

$$\phi_1, \ldots, \phi_l \vdash \psi \wedge \psi'$$

And suppose e realizes

$$\phi_1, \ldots, \phi_l, \phi \vdash \psi$$

Then $\Lambda m_1, \ldots, m_l.(\Lambda m.\{e\}(m_1, \ldots, m_l, m))$ realizes

$$\phi_1, \ldots, \phi_l \vdash \phi \rightarrow \psi$$

Conversely, suppose e realizes

$$\phi_1, \ldots, \phi_l \vdash \phi \rightarrow \psi$$

and e' realizes

$$\phi_1, \ldots, \phi_l \vdash \phi$$

Then $\Lambda m_1, \ldots, m_l.\{\{e\}(m_1, \ldots, m_l)\}(\{e'\}(m_1, \ldots, m_l))$ realizes

$$\phi_1, \ldots, \phi_l \vdash \psi$$

Finally, suppose e realizes

$$\phi_1, \ldots, \phi_l \vdash \psi(t)$$

where t is a closed term that evaluates to a number k, say. Then $\Lambda m_1, \ldots, m_l.\langle k, \{e\}(m_1, \ldots, m_l)\rangle$ realizes

$$\phi_1, \ldots, \phi_l \vdash \exists x.\, \psi(x)$$

(This is somewhat more complicated in the presence of free variables.) ■

The converse is not true: there are realizable formulas that are not derivable. We will see some examples presently. In fact, as Dummett has pointed out, from a philosophical point of view realizability really does not provide an adequate interpretation of constructive logic at all—even if you accept the realizable principles that we're about to list, and the other (less interesting) realizable formulas that are not intuitively 'constructive'.

Some 'constructive' principles that are realizable

If we extend realizability to higher-order logic (which we will do eventually) then we can say, roughly, that a formula ϕ is realizable exactly when it is a theorem of recursive arithmetic. In particular Church's Thesis, Markov's Principle and various constructively acceptable axioms of choice (countable choice, dependent choices etc.) are realizable. Even in our first-order setting it is possible to illustrate some of these.

Markov's Principle This is the following schema:

$$(\forall x.\, (\phi(x) \vee \neg\phi(x)) \wedge \neg\neg\exists x.\, \phi(x)) \longrightarrow \exists x. \phi(x)$$

Intuitively, the idea is as follows: suppose $\phi(x)$ is *decidable*, so that we have an effective way to test, for any $n \in \mathbb{N}$, whether $\phi(n)$ holds. And suppose that $\neg\neg(\exists n.\, \phi(x))$, i.e. that it is ridiculous to suppose that $\phi(n)$ fails for all n. Then we can simply run through each possibility $n = 0, 1, 2, \ldots$ in turn, until we find an n such that $\phi(n)$.

This seems plausible, though some constructivists would object to it as we cannot give a bound, in advance, of how long the search will take—that is, we do not have a 'constructive' way of telling when our search will succeed, but only know that it cannot fail. Anyway, it turns out to be realizable (though not *constructively* realizable, as we'll see).

We'll need to use the following fact, which roughly speaking says that the realizability interpretation is not interesting for doubly negated formulas (because such formulas have 'no computational content'). It is closely related to Gödel's double negation translation. This phenomenon will pop up again later on—it turns out to be central in the theory of $\mathcal{E}\!f\!f$.

Proposition 4.1.11 *If ϕ is realizable, any $n \in \mathbb{N}$ realizes $\neg\neg\phi$. If not, $\neg\neg\phi$ is not realizable.*

Proof. If ϕ is realizable, say by $k \in \mathbb{N}$, no $e \in \mathbb{N}$ can realize $\neg\phi = \phi \to \bot$ (otherwise $\{e\}(k)$ would realize $\bot$, and nothing can). So the condition on $n \in \mathbb{N}$ that it realize $\neg\neg\phi = (\neg\phi) \to \bot$ is vacuous. Therefore any $n \in \mathbb{N}$ realizes $\neg\neg\phi$.

Conversely, suppose ϕ is not realizable. Then anything realizes $\neg\phi$, so there can be nothing which realizes $\neg\neg\phi$. ■

Exercise 4.1.12 "If $\vdash \phi \vee \neg\phi$, then $\phi \leftrightarrow \neg\neg\phi$ and so we may as well assume that if ϕ is realizable, anything realizes ϕ." Make this precise and prove it. ■

Proposition 4.1.13 *Each instance of Markov's Principle is realizable.*

Proof. Markov's Principle in the *metatheory* will tell us that we can construct a realizer. For simplicity we assume that ϕ has no additional free variables.

Suppose that e realizes

$$(\forall x.\, (\phi(x) \vee \neg(\phi(x)))) \wedge \neg\neg(\exists x.\, \phi(x))$$

Then for all x,

$$\{e_0\}(x) = \begin{cases} \langle 0, n\rangle & n \text{ realizes } \phi(x) \\ \langle 1, n\rangle & n \text{ realizes } \neg\phi(x) \end{cases}$$

Assume that the first component $\{e_0\}(k)_0 = 1$ for all $k \in \mathbb{N}$. Then $\Lambda x.\, \{e_0\}(x)_1$ would realize $\forall x.\, \neg\phi(x)$.

On the other hand, e_1 realizes $\neg\neg(\exists x.\, \phi(x))$, which is inconsistent with $\forall x.\, \neg\phi(x)$. By exercise 4.1.9, we have a contradiction. Therefore for some k, $\{e_0\}(k)_0 = 0$, and $\{e_0\}(k)_1$ realizes $\phi(k)$. Therefore $\langle k, \{e_0\}(k)_1\rangle$ realizes $\exists x.\, \phi(x)$.

So $\Lambda e.\, \langle \mu k.(\{e_0\}(k)_0 = 1), \{e_0\}(\mu k.(\{e_0\}(k)_0 = 1)_1)\rangle$ realizes Markov's Principle. ■

Church's Thesis This one is a little harder. To state it, we need to recall some more definitions from elementary recursion theory.

Definition 4.1.14 *Kleene's T-**predicate** is a primitive recursive predicate such that*

$$T(e, n, y) \quad \textit{iff} \quad \text{“}y \textit{ is a code for a terminating computation of } \{e\}(n)\text{”}$$

*His **output function** is a primitive recursive function U such that*

$$T(e, n, y) \quad \textit{implies} \quad U(y) = \{e\}(n)$$

Troelstra showed that all this recursion theory can be formalised in our system of arithmetic. But for simplicity, we will assume that T has been added explicitly to the system, with suitable axioms, and that anything realizes T (since T is decidable, this is harmless: see 4.1.12). We will also assume that U has been added as a new function symbol. This simply avoids a lot of messiness in writing out the proof below.

Definition 4.1.15 ***Church's Thesis** is the following schema:*

$$\forall x.\exists z.\, \phi(x, z) \to \exists e.\forall x.\exists y.\, T(e, x, y) \wedge \phi(x, U(y))$$

We will show that this is realizable. In the following proof, if a number n is constructed using a number x (say) then we will indicate this dependency by writing n_x. Similarly we write $n_{x,y}$ if n depends on both x and y. I hope this doesn't cause too much confusion.

Proposition 4.1.16 *Any instance of Church's Thesis is realizable.*

Proof. We need to find a code $r \in \mathbb{N}$ such that if n realizes $\forall x.\exists z.\, \phi(x, z)$ then $\{r\}(n)$ realizes

$$\exists e.\forall x.\exists y.\, T(e, x, y) \wedge \phi(x, U(y))$$

So suppose that n realizes

$$\forall x.\exists z.\, \phi(x, z)$$

This means that for all x, $\{n\}(x)$ realizes $\exists z.\, \phi(x, z)$; that is, $\{n\}(x) = \langle k_x, m_x \rangle$ where m_x realizes $\phi(x, k_x)$.

Let $e_n = \Lambda x.\{n\}(x)_0$. Then for all x, $k_{n,x} = \{e_n\}(x)$ is defined; in particular $T(e_n, x, y_{n,x})$ holds, for some smallest $y_{n,x} \in \mathbb{N}$. Picking some arbitrary $p \in \mathbb{N}$, then, $q_{n,x} = \langle p, \{n\}(x)_1 \rangle$ realizes $T(e, x, y_{n,x}) \wedge \phi(x, U(y_{n,x}))$ since $k_{n,x} = \{e_n\}(x) = U(y_{n,x})$; i.e. $\{n\}(x)_1$ is the m_x mentioned above. (Any p will do because of the remark we made about primitive recursive predicates.) So $f_{n,x} = \langle y_{n,x}, q_{n,x} \rangle$ realizes

$$\exists y.\, T(e_n, x, y) \wedge \phi(x, U(y))$$

and hence $\Lambda x.f_{n,x}$ realizes

$$\forall x.\exists y.\, T(e_n, x, y) \wedge \phi(x, U(y))$$

So we can let $r = \Lambda n.\langle e_n, \Lambda x.f_{n,x} \rangle$. ∎

There is actually a higher-order version of Church's Thesis which we can state in the internal language of a topos and which will also turn out to be valid in the effective topos. It is

$$\forall f : N^N.\, \exists e : N.\, \forall n : N.\, \exists y : N.\, T(e, n, y) \wedge f(n) = U(y)$$

As Lifschitz showed, this is *not* equivalent to the first-order schema, but is strictly weaker. Jaap van Oosten has found a topos-theoretic version of the proof.

Exercise 4.1.17 Extended Church's Thesis is the schema

$$\forall x.\,(\phi(x) \to \exists y.\psi(x,y)) \;\to\; \exists e.\forall x.\,\phi(x) \to \exists y.\,T(e,x,y) \wedge \psi(x,U(y))$$

where ϕ does not contain $\vee$, and contains $\exists$'s only in front of atomic formulas $t = t'$. Show that this is always realizable. Troelstra used this principle to characterise those formulas which were *provably* realizable. Can you think of a higher-order version of extended Church's Thesis? ■

The problem, then, is to extend Kleene's interpretation to constructive set theory. Hyland did just that, by constructing a suitable topos. In fact he was anticipated by Powell, who worked within the framework of intuitionistic ZF. They both started with roughly the same idea, namely to generalise the notion of boolean-valued model.

We will sketch Hyland's construction in section 4.4. First, we trace another line of development that also leads to the effective topos: namely, the search for 'set-theoretic' models of polymorphism.

4.2 The PER model

Back in the '60s, Strachey discussed different notions of polymorphism. In particular he distinguished between *ad hoc polymorphism* (different functions which have the same name but are implemented separately: e.g. addition for integers and reals) and *parametric polymorphism* (a family of functions defined for all datatypes and all implemented using the same piece of code: e.g. reversing lists).

Reynolds seems to have been the first to give some formal substance to Strachey's idea of parametric polymorphism, by coming up with the second order λ-calculus. This actually turned out to be[5] Girard's System F, invented some years earlier for metamathematical purposes. Anyway, it was at first not clear whether the system had a semantics: for example, though the simply typed λ-calculus has a simple set-theoretic semantics, Reynolds showed that this was not possible for System F.

We will sketch a recursion-theoretic semantics for System F in terms of *partial equivalence relations* on $\mathbb{N}$. Generalising this semantics to cope with more complicated type theories, attempting to deal with phenomena such as equality types and trying to salvage something from Reynolds' idea that there ought to be a simple set-theoretic semantics, will lead us to ω-sets and to the effective topos.

Semantics using PERs

Recall that the types of System F are

$$\tau \quad ::= \quad X \mid \tau \to \tau' \mid \Pi X.\tau$$

Reynolds' idea was to model this using some collection $\mathbf{C}$ of sets, the denotations of closed types. Arrow types would be interpreted using set-theoretic function spaces and polymorphic types might, for example, be modelled simply as products (an idea associated with Moggi and Hyland):

$$[\![\Pi X.X \to X]\!] = \prod_{C \in \mathbf{C}} C^C$$

[5]—a subsystem of—

Actually, this simple-minded idea seems unlikely to work. In fact, Reynolds later showed that there could be no set-theoretic model, even if we don't try to use products to model Π-types but look for some cleverer construction.

We will now construct a cartesian closed category of partial equivalence relations on ℕ, and show that it provides a model of System F in which polymorphic types are modelled using intersections. We will then explain Moggi's insight that these are really products after all, in a suitable sense; and by the end of these notes, it will transpire that this really is a set-theoretic model, though it doesn't seem like one at first.

Definition 4.2.1 *A* ***partial equivalence relation*** *or* ***PER*** *on* $\mathbb{N}$ *is a symmetric, transitive relation* $R \subseteq \mathbb{N} \times \mathbb{N}$*. The* ***domain*** *of* R *is*

$$\mathrm{Dom}\, R = \{n \in \mathbb{N} | \, nRn\} \subseteq \mathbb{N}$$

Example 4.2.2 Define a PER N by letting nNm iff $n = m$; i.e. N is the equality relation on $\mathbb{N}$. This is the PER which plays the role of $\mathbb{N}$ itself. We will see later on that N is a natural numbers object in the category of PERs, and in fact in $\mathcal{E}\!f\!f$. ■

Example 4.2.3 Define a PER 1 by letting $n1m$ for all $n, m \in \mathbb{N}$. This will turn out to be a terminal object in the category of PERs. Of course, there will also be plenty of other ways to describe the terminal object. ■

Definition 4.2.4 *Let* R, S *be PERs. A code* $e \in \mathbb{N}$ ***tracks an arrow*** $R \to S$ *if*

$$nRn' \text{ implies } \{e\}(n) S \{e\}(n')$$

Suppose e, e' *track arrows* $R \to S$*. Then* $e \sim e'$ *if*

$$n \in \mathrm{Dom}\, R \text{ implies } \{e\}(n) S \{e'\}(n)$$

(Note that the definition of $e \sim e'$ *is relative to the PERs* R, S*.)*

Definition 4.2.5 *An* ***arrow*** $f \colon R \to S$ *is an equivalence class of codes tracking arrows* $R \to S$*. We sometimes write* $f = [e]$ *where* $e \in \mathbb{N}$ *is a representative for* f*. We let* Per *denote the category of PERs and maps between them.*

We'll see in the next section that arrows between PERs can be described in a rather more intuitive way. For the moment, though, let's play with the definition we've got.

Proposition 4.2.6 Per *is a cartesian closed category.*

Proof. We will give the constructions and leave the verifications as exercises. If R, S are PERs put

$$n(R \times S)m \qquad \text{iff} \qquad n_0 R m_0 \text{ and } n_1 S m_1$$

There are projection maps $R \times S \to R, R \times S \to S$ tracked by $\Lambda n.n_0, \Lambda n.n_1$.

We also define a PER S^R as follows:

$$\mathrm{Dom}\, S^R = \{e \in \mathbb{N} | \, e \text{ tracks an arrow } R \to S\}$$

and setting $e(S^R)e'$ iff $e \sim e'$. The evaluation arrow

$$S^R \times R \to S$$

is tracked by $\Lambda n.\{n_0\}(n_1)$. ■

Example 4.2.7 We have

$$\text{Dom } N^N = \{e \in \mathbb{N} |\, e \text{ is an code for a (total) recursive function}\}$$

with $e(N^N)e'$ iff e and e' are codes for the same recursive function. ■

Exercise 4.2.8 Show that Per has equalizers: if

$$[e_1], [e_2]: R \rightrightarrows S$$

let R' be the PER with domain

$$\text{Dom } R' = \{n \in \text{Dom } R |\, \{e_1\}(n) S \{e_2\}(n)\}$$

and put, for $n, m \in \text{Dom } R'$, $nR'm$ iff nRm. There is an inclusion arrow $R' \hookrightarrow R$, tracked by $\Lambda n.n$. ■

Exercise 4.2.9 Show that Per has binary coproducts: put

$$\text{Dom } R + S = \{\langle 0, n\rangle |\, n \in \text{Dom } R\} \cup \{\langle 1, m\rangle |\, m \in \text{Dom } S\}$$

Does it have countable coproducts? ■

Anyway, 4.2.6 tells us that Per can be used to model the simply typed λ-calculus. It turns out that it can also be used to model System F: polymorphic types are modelled as *intersections* in the obvious way. For example, we can let

$$[\![\Pi X.X \to X]\!] = \bigcap_{R \text{ a PER}} R^R$$

For any PER R, the identity arrow (tracked by $\Lambda n.n$) gives rise to an arrow

$$[\Lambda k.(\Lambda n.n)]: 1 \to R^R$$

So we get a map

$$[\Lambda k.(\Lambda n.n)]: 1 \to \bigcap_{R \text{ a PER}} R^R$$

which is just the interpretation $[\![\Lambda X.\lambda x^X.x]\!]$ of the polymorphic identity.

PERs as a fibration over $\mathcal{S}ets$

We won't describe in detail how to use Per to interpret System F. All this material is more than adequately dealt with in thousands of papers written over the course of centuries. However, we'll describe a suitable fibration of PERs over $\mathcal{S}ets$ that is a 'model of System F' in the sense described in example 2.4.3.

Definition 4.2.10 *Define a fibration* $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{array}$ *as follows:* $\mathbb{P}_I$ *is the category with*

objects: *families* $\{R_i\}_{i \in I}$ *where each* R_i *is a PER*

arrows: *an arrow* $\{R_i\}_{i\in I} \to \{S_i\}_{i\in I}$ *is given by an equivalence classes of codes* $e \in \mathbb{N}$ *such that for each* $i \in I$, e *tracks an arrow* $R_i \to S_i$

where e, e' *are equivalent if for all* $i \in I$, $e \sim e'$ *as codes for arrows* $R_i \rightrightarrows S_i$.

Note the *uniform* definition of morphism; this is what makes things work. Anyway, Moggi was the first to notice the following crucial facts, which say that the polymorphic types we calculated before are products after all—at least in a fibred sense.

Proposition 4.2.11 $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{array}$ *is a complete fibration.*

Proof. Let I be a set; we will define a right adjoint Π_I to the reindexing functor $I^*: \mathbb{P}_1 \cong \mathsf{Per} \to \mathbb{P}_I$. In fact we simply let

$$\Pi_I(\{R_i\}_{i\in I}) = \bigcap_{i\in I} R_i$$

We need to show that if S is a PER, there is a natural bijection

$$\frac{I^*S \to \{R_i\}_{i\in I}}{S \to \Pi_I(\{R_i\}_{i\in I})}$$

A representative for an arrow $I^*S \to \{R_i\}_{i\in I}$ in $\mathbb{P}_I$ is a code $e \in \mathbb{N}$ such that e tracks a map $S \to R_i$ for each $i \in I$; i.e.

$$nSn' \text{ implies } \{e\}(n)R_i\{e\}(n') \text{ for all } i \in I$$

This is exactly the condition that e track an arrow $S \to \bigcap_{i\in I} R_i = \Pi_I(\{R_i\}_{i\in I})$. The converse is also easy once we notice that $\Lambda n.n$ tracks a map

$$\Pi_I(\{R_i\}_{i\in I}) = \bigcap_{i\in I} R_i \to R_i$$

for each $i \in I$ (this is the counit of the adjunction).

Now suppose, more generally, that we have $\phi: J \to I$ and we want to define a right adjoint Π_ϕ to $\phi^*: \mathbb{P}_J \to \mathbb{P}_I$. We let

$$\Pi_\phi(\{R_j\}_{j\in J}) = \{\bigcap_{\phi(j)=i} R_j\}_{i\in I}$$

The rest of the details are left to the reader. Note that the Beck-Chevalley condition trivially holds. ■

Example 4.2.12 Show that $\begin{array}{c}Fam(\mathsf{Per})\\ \downarrow p\\ \mathcal{S}ets\end{array}$ is not complete. That is, if we turn Per into a fibration over $\mathcal{S}ets$ in the usual way, we do not obtain a complete fibration: the uniformity in the definition of morphism is essential. ■

Proposition 4.2.13 $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{array}$ *has a generic object* $G \in \mathbb{P}_{P_0}$.

Proof. We let P_0 be the set of PERs on $\mathbb{N}$, and $G = \{R\}_{R\in P_0}$. ■

Exercise 4.2.14 Is this a skeletal generic object? ■

So we can model polymorphic types in our fibration in the usual way. For example,

$$[\![\Pi X.X \to (X \to X)]\!] = \Pi_{P_0}(G^G)^G$$

But we can't do everything we'd like. For example, though we can quantify over all 'types', we can't quantify over 'equality types' (*á la* ML), as the following example shows.

Example 4.2.15 Call a PER R **decidable** if there is a code $e \in \mathbb{N}$ such that for all $n, n' \in \mathrm{Dom}\, R$,

$$\{e\}(n, n') = 0 \text{ or } 1, \text{ and } \{e\}(n, n') = 1 \text{ iff } nRn'$$

(That is, if R-equivalence is effectively decidable; we call e a **decision code** for R. See 4.3.8 for more of an explanation.) It's easy to see that R is decidable iff there exists a 'suitable' arrow $R \times R \to 2 = 1 + 1$—an 'equality function'. (If e is a decision code for R then this arrow is tracked by, say, $\Lambda n.\{e\}(n_0, n_1)$.)

More generally, call a family $\{R_i\}_{i \in I}$ of PERs **decidable** if there is a single code $e \in \mathbb{N}$ such that for all $i \in I$ and for all $n, n' \in \mathrm{Dom}\, R_i$,

$$\{e\}(n, n') = 0 \text{ or } 1, \text{ and } \{e\}(n, n') = 1 \text{ iff } nR_in'$$

Such families are the objects of a full subfibration of $\begin{smallmatrix}\mathbb{P}\\ \downarrow \\ \mathcal{S}ets\end{smallmatrix}$.

This fibration does not have a generic object. To see why, consider the only candidate for such a generic object: the family $\{R\}_{R \in E_0}$ where E_0 is the set of decidable PERs. There is no reason why a single code $e \in \mathbb{N}$ should track a suitable map $R \times R \to 2$, for all decidable R. ■

In defining what a decidable *family* of PERs was, we imposed the 'uniformity' condition so that decidability would still be equivalent to the existence of a map $\{R_i\}_{i \in I} \to I^*2$. To see why this is desirable, suppose we are reasoning about an (indeterminate) equality type X in, say, ML. We would certainly expect there to be a function

$$\mathrm{eq}\colon X \times X \to \mathrm{Bool}$$

and this would be interpreted, in our fibration, by an arrow $G' \times G' \to 2$ where G' is the generic family of decidable PERs. The fact that no such thing exists is clearly a drawback.

Intuitively, decidability is not a 'uniform' property: different PERs can be decidable for different reasons. But we will see in 4.3.15 that this need not stop us from constructing a fibration of decidable PERs which does have a generic object. The key is to allow indexing objects which are not just sets, but contain intensional information about the families they index: in this case, codes for decision algorithms.

Modelling type dependency

Let us explain another problem with considering families of PERs indexed only by ordinary sets. Ever since Martin-Löf type theory became fashionable, everyone's been talking about dependent types like

$$\tau(x)\,[x : \sigma]$$

and form dependent products $\Pi x : \sigma.\tau(x)$ or sums $\Sigma x : \sigma.\tau(x)$. In a PER model, $[\![\tau(x)]\!]$ would be a family of PERs indexed by the PER $[\![\sigma]\!]$. But we have not defined any such notion. And in fact, the fibration $\begin{smallmatrix}\mathbb{P}\\ \downarrow \\ \mathcal{S}ets\end{smallmatrix}$ is useless for modelling dependent types, for the following technical reason:

Exercise 4.2.16 Per, fibred over $\mathcal{S}ets$, is not locally small. (Hint: use the same method as in example 2.4.11.) ∎

The intuition is quite simple: if R and S are PERs, there is no *set* which can parametrise the maps $R \to S$ properly—this is because these maps really form a *PER*, namely S^R, not just a set.

Exercise 4.2.17 Show that $\begin{smallmatrix} Fam(\mathsf{Per}) \\ \downarrow p \\ \mathcal{S}ets \end{smallmatrix}$ —i.e. Per, fibred over $\mathcal{S}ets$ in the 'incorrect' way—*is* locally small. So if we insist on using $\mathcal{S}ets$ as our base category, then we can have a locally small fibration or a complete fibration but not both. ∎

The only way to fix the situation is to modify the base category: to use some bigger category than $\mathcal{S}ets$, over which PERs can form a complete and locally small fibration. The right base category turns out to be the category of ω-sets.

Before explaining this, we show that PERs can be fibred over *themselves* to obtain a locally small fibration and hence a model of dependent type theory.

Families of PERs indexed by a PER Recall that any category with pullbacks can be fibred over itself: see 2.2.9. In the case of Per, this gives us a fibration

$$\begin{matrix} \mathsf{Per}^{\to} \\ \downarrow \\ \mathsf{Per} \end{matrix}$$

Proposition 4.2.18 *This fibration is complete. In fact,* Per *is locally cartesian closed.*

Proof. Suppose Q, R, S are PERs, $[m]: Q \to S, [n]: R \to S$. We will construct the exponential $\begin{pmatrix} R \\ \downarrow \\ S \end{pmatrix}^{\begin{pmatrix} Q \\ \downarrow \\ S \end{pmatrix}}$ in Per/S. The reader can check that the construction actually works[6].

We wish to define a PER T and an arrow $T \to S$. First, if $k \in \mathrm{Dom}\, S$ let Q_k be the following PER:

$$\mathrm{Dom}\, Q_k = \{p \in \mathrm{Dom}\, Q |\, \{m\}(p) S k\}, \qquad pQ_k q \text{ iff } pQq$$

Note that if kSk' then $Q_k = Q_{k'}$.

Now define T as follows: let

$$\mathrm{Dom}\, T \quad = \quad \{\text{pairs } \langle k, e \rangle, \text{ where } k \in \mathrm{Dom}\, S \text{ and } e \text{ tracks an arrow } Q_k \to R_k\}$$

and let $\langle k, e\rangle T \langle k', e' \rangle$ iff kSk' and $e \sim e'$ as codes for arrows $Q_k \to R_k$. There is an obvious arrow $T \to S$ tracked by $\Lambda n.n_0$. ∎

So Per provides us with a model of dependent type theory. However, notice that this does not give us a model of polymorphism as there will be no generic object (since the set of all PERs is not a PER). So it does not solve the problem of combining dependent types with polymorphism. Also, calculating with PERs like this is a little inconvenient, and it is not clear what an object of Per/S has to do with an honest-to-goodness family of PERs—though the proof gives us some indication. We will take this up in exercise 4.3.19.

Note that local smallness follows from local cartesian closure: see 2.4.14.

[6] If the reader can be bothered; it's messy. The proposition follows from more general results to be described later, that give a unified account of the different completeness properties of Per.

4.3 The category of ω-sets

It seems natural[7] to look for a kind of common generalisation of PERs and ordinary sets, i.e. to find a single category which contains both Per and *Sets*. We might then hope that this category, fibred over itself, would contain a full subfibration of 'families' of PERs that supported an interpretation of dependent types and polymorphism. We would expect to be able to recover our original fibration $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{array}$ by change of base.

The following definition appears, slightly disguised, in [7, section 6] and seems to date back to ideas of Ershov, in particular the notion of 'enumerated set'. More information, in particular on computer science applications, appears in the more recent papers [15] and [4].

Definition 4.3.1 *An **ω-set** or **$P\mathbb{N}$-valued assembly** $(X, \in)$ consists of a set X together with, for each $x \in X$, a subset*

$$\varnothing \neq |x \in X| \subseteq \mathbb{N}$$

*An **arrow** $f\colon (X, \in) \to (Y, \in)$ between two assemblies is a function $f\colon X \to Y$ between their underlying sets, for which there exists a code $e \in \mathbb{N}$ such that for all $x \in X$,*

$$\textit{if } n \in |x \in X| \textit{ then } \{e\}(n) \in |f(x) \in Y|$$

*e is said to **track** f. We denote the category of ω-sets and arrows by ω-$\mathcal{S}ets$.*

Note that the definition of arrow is 'extensional': if different codes $e, e' \in \mathbb{N}$ both track f then they still describe a single arrow f, not two distinct arrows.

Proposition 4.3.2 *ω-$\mathcal{S}ets$ is a cartesian closed category.*

Proof. Let $(X, \in), (Y, \in)$ be ω-sets. $(X, \in) \times (Y, \in)$ has underlying set $X \times Y$ and

$$|\langle x, y\rangle \in X \times Y| = \{\langle n, m\rangle |\, n \in |x \in X| \text{ and } m \in |y \in Y|\}$$

$(Y, \in)^{(X, \in)}$ has underlying set

$$Z = \{\text{arrows } f\colon (X, \in) \to (Y, \in)\}$$

with

$$|f \in Z| = \{e \in \mathbb{N} |\, e \text{ tracks } f\}$$

The verifications are easy. ∎

In fact, ω-$\mathcal{S}ets$ is actually a locally cartesian closed category, and hence supports an interpretation of dependent type theory as well as the simply typed λ-calculus. It is not hard, though somewhat messy, to prove this directly. But we won't bother, since this fact follows immediately from its logical characterisation as a full subcategory of $\mathcal{E}\!f\!f$ and some general topos theory. See section 4.5.

Definition 4.3.3 *N is the ω-set $(\mathbb{N}, \in)$ where*

$$|n \in \mathbb{N}| = \{n\}$$

(Cf. example 4.2.2.)

[7] in hindsight

Proposition 4.3.4 *N is a natural numbers object in ω-$\mathcal{S}ets$.*

Proof. First note that we do have arrows

$$1 \xrightarrow{0} N \xrightarrow{s} N$$

0 is tracked by $\Lambda k.0$, and s is tracked by $\Lambda n.n+1$.

Now suppose we have an ω-set $(X, \in)$, together with arrows

$$1 \xrightarrow{f} (X, \in) \xrightarrow{g} (X, \in)$$

We want to define $h: N \to (X, \in)$ such that the following diagram commutes:

$$\begin{tikzcd}
 & N \arrow[r, "s"] \arrow[d, "h"] & N \arrow[d, "h"] \\
1 \arrow[ur, "0"] \arrow[dr, "f"'] & & \\
 & (X, \in) \arrow[r, "g"'] & (X, \in)
\end{tikzcd}$$

Note that there is an obvious function $\mathbb{N} \to X$ (taking $n \mapsto g^n(f(*))$) which must be the underlying function of h. So h is uniquely determined if it exists at all. To show that h exists, we must find a code that tracks it.

But suppose $k \in \mathbb{N}$ tracks f and $e \in \mathbb{N}$ tracks g. Then there is a code $e' \in \mathbb{N}$ for the following total recursive function:

$$\{e'\}(n) = \begin{cases} k & n = 0 \\ \{e\}(\{e'\}(n-1)) & n > 0 \end{cases}$$

Clearly e' tracks h. ■

Ordinary sets as ω-sets There is a full embedding

$$\nabla: \mathcal{S}ets \hookrightarrow \omega\text{-}\mathcal{S}ets$$

letting us regard any set X as an ω-set ∇X. We simply put

$$|x \in X| = \mathbb{N}$$

It is easy to check that if $f: X \to Y$ is a set function, any code for a total recursive function (e.g. $\Lambda x.x$) tracks an arrow $\nabla f: \nabla X \to \nabla Y$ in ω-$\mathcal{S}ets$.

Exercise 4.3.5 Suppose $(X, \in) \in \omega$-$\mathcal{S}ets$ and $Y \in \mathcal{S}ets$. If $f: X \to Y$ is any set function, there is an arrow $\bar{f}: (X, \in) \to \nabla Y$ whose underlying function is f; furthermore $\bar{f}$ factors uniquely through ∇f. ■

So $\mathcal{S}ets$ actually turns out to be a reflective subcategory of ω-$\mathcal{S}ets$. Although this is not hard to prove directly, the best way to understand the result is in the topos-theoretic setting we'll describe later. In particular, the embedding has various logical properties, generalising facts about realizability (e.g. telling us which formulas are 'self-realizing') which are best understood topos-theoretically. However, these logical properties won't be discussed in these notes (except in 4.5.25).

PERs as ω-sets

We will now construct a full embedding of Per into ω-$\mathcal{S}ets$ and characterize those ω-sets which are isomorphic to PERs under this embedding. This ties down the sense in which ω-sets are a common generalisation of PERs and ordinary sets.

Proposition 4.3.6 *There is a full embedding* $\mathsf{Per} \hookrightarrow \omega$-$\mathcal{S}ets$.

Proof. Let R be a PER, and let $\mathbb{N}/R$ be its quotient. That is,

$$\mathbb{N}/R = \{p \subseteq \mathbb{N}|, p \text{ is an } R\text{-equivalence class}\}$$

Define

$$|p \in \mathbb{N}/R| = p$$

Then $(\mathbb{N}/R, \in)$ is an ω-set. Furthermore, if $e \in \mathbb{N}$ tracks a map $R \to S$, then the map $\mathbb{N}/R \to \mathbb{N}/S$ given by

$$p \mapsto (\text{the } S\text{-equivalence class of } \{e\}(n), \text{ for some } n \in p)$$

is well defined. So we obtain a map $(\mathbb{N}/R, \in) \to (\mathbb{N}/S, \in)$ tracked by e. Faithfulness and fullness are both easy to verify. ■

Note that the PER N defined in example 4.2.2 is identified with the ω-set N under this embedding. It follows that the PER N is a natural numbers object in Per.

Exercise 4.3.7 Call an ω-set $(X, \in)$ **effective** if for all $x, x' \in X$

$$\text{if } x \neq x' \text{ then } |x \in X| \cap |x' \in X| = \varnothing$$

Show that $(X, \in)$ is effective exactly when it is isomorphic to $(\mathbb{N}/R, \in)$, for some PER R. ■

Exercise 4.3.8 Let 2 be the ω-set with underlying set $\{\mathit{tt}, \mathit{ff}\}$ and

$$|\mathit{tt} \in 2| = \{0\}, \qquad |\mathit{ff} \in 2| = \{1\}$$

(i.e. 2=1+1). Call an ω-set $(X, \in)$ **decidable** iff there's an arrow

$$e: (X, \in) \times (X, \in) \to 2$$

such that $e(x, x') = \mathit{tt}$ iff $x = x'$. Show that if $(X, \in)$ is decidable, it's effective. Then show that $(X, \in)$ is decidable iff the corresponding PER (as in the previous exercise) is decidable in the sense of exercise 4.2.15. See also exercise 4.5.10. ■

Proposition 4.3.9 Per *is reflective in* ω-$\mathcal{S}ets$.

Proof. Given any ω-set $(X, \in)$, define $\bar{X} = X/\sim$ where

$$x \sim x' \quad \text{iff} \quad |x \in X| \cap |x' \in X| \neq \varnothing$$

Then define, for $\bar{x} \in \bar{X}$,

$$|\bar{x} \in \bar{X}| = \bigcup_{x \in \bar{x}} |x \in X|$$

■

This proposition is no use to us, actually. We cannot deduce useful completeness properties from it; we can only do that if the proposition is true in a suitable 'indexed' or 'fibred' sense. Our next task, then, is to see how Per can be regarded as a fibration over ω-$\mathcal{S}ets$. But we first mention a strong completeness property of Per which is a taste of things to come.

Exercise 4.3.10 The inclusion $\mathsf{Per} \hookrightarrow \omega\text{-}\mathcal{S}ets$ preserves exponentials. ■

Exercise 4.3.11 Per is an **exponential ideal** in $\omega\text{-}\mathcal{S}ets$: if $(X, \in)$ is an effective ω-set and $(Y, \in)$ is any old ω-set, then $(X, \in)^{(Y,\in)}$ is again an effective ω-set. ■

This property, together with closure under products indexed by ordinary sets (as in the last section) actually implies closure under products indexed by arbitrary ω-sets. We will not follow up this line of thought, but see [8].

Exercise 4.3.12 Let $X \in \mathcal{S}ets$ and let $(Y, \in)$ be effective. Then $(Y, \in)^{\nabla X} \cong (Y, \in)$. ■

PERs as a fibration over ω-$\mathcal{S}ets$

Definition 4.3.13 *The fibration* $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$ *is defined as follows. Let* $\mathbb{P}_{(X,\in)}$ *be the following category:*

objects: *families* $\{R_x\}_{x \in X}$ *where each* R_x *is a PER*

arrows: *an arrow* $\{R_x\}_{x\in X} \to \{S_x\}_{x \in X}$ *is given by an equivalence classes of codes* $e \in \mathbb{N}$ *such that for each* $x \in X$ *and* $n \in |x \in X|$, $\{e\}(n)$ *tracks an arrow* $R_x \to S_x$

We say that e ***witnesses*** *the arrow* $\{R_x\}_{x\in X} \to \{S_x\}_{x\in X}$. *The equivalence relation is defined in the obvious way, cf. definition 4.2.10.*

Note that $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{smallmatrix}$ may be obtained from $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$ via change of base along $\mathcal{S}ets \hookrightarrow \omega\text{-}\mathcal{S}ets$, so we have really just souped up our original fibration. The new feature here is the definition of morphism between families: we no longer require a code e that, for each x, tracks an arrow

$$R_x \to S_x$$

The witness e is allowed to look at $n \in |x \in X|$ to compute a code for such an arrow.

Proposition 4.3.14 $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$ *is a complete fibration with a generic object.*

The generic object is the same as for $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{smallmatrix}$ (essentially; it is indexed by ∇P_0 this time). We could prove completeness directly, as in [8], but this would take a long time. It actually follows from the fact that the modest sets form a complete full subcategory in $\mathcal{E}\!f\!f$ (in a suitable fibrational sense); and the proof of *that* fact is rather nice, though not very short either.

Exercise 4.3.15 Let's see what we've gained, by considering decidable PERs again (see 4.2.15). Consider the full subfibration of $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$ whose objects are the families of decidable PERs. Show that this *does* have a generic object. Hint: let $(D_0, \in)$ be the ω-set with underlying set

$$D_0 = \text{set of decidable PERs } R$$

and

$$|R \in D_0| = \text{set of decision codes for } R$$

and let the generic object be the family $\{R\}_{R\in D_0}$. Note that this time, we *do* have an arrow

$$\{R\}_{R\in D_0} \to \{2\}_{R\in D_0} \in \mathbb{P}_{(D_0,\in)}$$

The code $\Lambda e.\Lambda n.\langle\{e\}(n),42\rangle$ works; it can 'look at the decision code e'. We'll see later that the definition of $(D_0,\in)$, and of the fibration of decidable PERs, can be carried out entirely in the internal language of $\mathcal{E}\!f\!f$: 4.6.14. That is not so amazing in this simple case (though we'll see it's still pretty useful) but it's a very convenient technique if we're interested in studying more complicated properties of PERs. ■

Families of PERs as display maps Recall from proposition 4.2.18 that Per was a locally cartesian closed category. This meant that we could use the fibration $\begin{smallmatrix}\mathsf{Per}^{\to}\\ \downarrow\\ \mathsf{Per}\end{smallmatrix}$ to model dependent type theory, thinking of an arrow $R\to S$ as displaying "a family of PERs indexed by a PER S".

Since ω-$\mathcal{S}ets$ has pullbacks, it makes sense to talk about the fibration $\begin{smallmatrix}\omega\text{-}\mathcal{S}ets^{\to}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$. This is a complete fibration because ω-$\mathcal{S}ets$ is locally cartesian closed—a fact which, as we noted earlier, follows immediately from the characterisation of ω-sets in section 4.5.

Lemma 4.3.16 *There is a fully faithful cartesian functor* $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix} \hookrightarrow \begin{smallmatrix}\omega\text{-}\mathcal{S}ets^{\to}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$.

Proof. Let $\{R_x\}_{x\in X}\in\mathbb{P}_{(X,\in)}$. Define an ω-set $(Y,\in)$ by letting Y be the disjoint union

$$Y = \coprod_{x\in X}\mathbb{N}/R_x$$

and if $\langle x,p\rangle\in Y$ (so that p is an R_x-equivalence class) putting

$$|\langle x,p\rangle\in Y| = \{\langle k,n\rangle \mid k\in|x\in X|, n\in p\}$$

There is an obvious map $(Y,\in)\to(X,\in)$ taking $\langle x,p\rangle\mapsto x$.

Suppose we have an arrow $\{R_x\}_{x\in X}\to\{S_x\}_{x\in X}$ witnessed by $e\in\mathbb{N}$. If $(Z,\in)$ is the ω-set constructed from $\{S_x\}_{x\in X}$ as above, $\Lambda r.\langle r_0,\{e\}(r_1)\rangle$ tracks the desired map $(Y,\in)\to(Z,\in)$ over $(X,\in)$.

Conversely, suppose we have an arrow $f\colon(Y,\in)\to(Z,\in)$ over $(X,\in)$, tracked by $e\in\mathbb{N}$. Then $\Lambda k.(\Lambda n.(\{e\}(n))_0)$ witnesses the corresponding arrow $\{R_x\}_{x\in X}\to\{S_x\}_{x\in X}$. ■

Proposition 4.3.17 $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$*, as a subfibration of* $\begin{smallmatrix}\omega\text{-}\mathcal{S}ets^{\to}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$*, is closed under* Π*-functors.*

Proof. A direct proof is possible (cf. [8]) but would be very tedious. The result will follow from an alternative description of $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$, given later, and a general result on 'modest sets'. ■

Corollary 4.3.18 *The cartesian inclusion*

$$\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix} \hookrightarrow \begin{smallmatrix}\omega\text{-}\mathcal{S}ets^{\to}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$$

gives rise to a model of the calculus of constructions, with strong sums.

We omit the proof, as we haven't defined what a model of the calculus of constructions is.

Exercise 4.3.19 Show that if we regard $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ as a full subfibration of $\begin{array}{c}\omega\text{-}\mathcal{S}ets^{\to}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$, we can obtain the fibration $\begin{array}{c}\mathsf{Per}^{\to}\\ \downarrow\\ \mathsf{Per}\end{array}$ from it by change of base along $\mathsf{Per} \hookrightarrow \omega\text{-}\mathcal{S}ets$. Deduce that $\begin{array}{c}\mathsf{Per}^{\to}\\ \downarrow\\ \mathsf{Per}\end{array}$ is a complete fibration, and that Per is locally cartesian closed. ■

The internal category of PERs The reader may be familiar with another categorical method of modelling polymorphic type theories, based on *internal categories* with suitable completeness properties. We sketch how the PER model fits into this setting.

Recall that P_0 was the set of PERs on $\mathbb{N}$. We abuse notation by writing P_0 for the ω-set ∇P_0. Also, recall that the notion of internal category makes sense in any category with finite limits; in particular it makes sense to talk about internal categories in ω-*Sets*.

Definition 4.3.20 *The internal category* $\mathbf{P} = (P_1 \rightrightarrows P_0)$ *in* ω-*Sets is defined as follows:* $P_1 = (Z, \in)$ *where* Z *is the disjoint union*

$$Z = \coprod_{R,S \in P_0} \{\textit{arrows } f\colon R \to S \textit{ in } \mathsf{Per}\}$$

and $|f \in Z| = \{e \in \mathbb{N} | \, e \textit{ tracks } f\}$.

The following facts can be proved:

- $\mathbf{P}$ is "complete relative to ω-*Sets*";
- $\mathbf{P}$ is an "internal full subcategory of ω-*Sets*"; and
- $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ is the "externalisation" of $\mathbf{P}$;

In particular, $\mathbf{P}$ gives us an 'internal' model of System F (and in fact System $F\omega$, among other things); and the interpretation of types and terms in this model is 'the same' as in the fibrational model, using $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$.

The precise definitions, and certainly the proofs, are rather technical. Later on we will see how to construct $\mathbf{P}$ as an internal category **Mod** in the topos $\mathcal{E}\!f\!f$, in an intuitive way: see 4.6.12. These statements can then be formulated set-theoretically using the internal language of $\mathcal{E}\!f\!f$.

The problem with ω-*Sets*

So, we have a model of polymorphism and dependent types which we would like to regard, somehow, as a 'set-theoretic' model; and we would like to do this in the framework of a model of constructive set theory which generalises realizability for constructive first-order arithmetic. Finally, we'd like to carry all this out in a categorical setting.

Why not use ω-*Sets*? It turns out to be a so-called *quasitopos*, and such things support a great deal of categorical logic. They can't, however, be regarded as models of constructive set theory. So if our eventual goal is to reason informally about objects in the model, using set-theoretic arguments, it is much safer to embed ω-*Sets* in a category which *is* a topos.

First, let's see what goes wrong with ω-*Sets*. The problem has to do with epimorphisms and equivalence relations, and one way of explaining it is as follows. Recall from 3.3.1 that every arrow in $\mathcal{E}$ which is both mono and epi is an isomorphism. This convenient fact fails in ω-*Sets*.

We can see this by first giving characterisations of when an arrow

$$f\colon (X, \in) \to (Y, \in) \in \omega\text{-}\mathcal{S}ets$$

is mono or epi in ω-*Sets*. As equality of arrows in ω-*Sets* is determined by equality of underlying functions, the following facts are straightforward. We prove one and leave the other as an exercise.

Exercise 4.3.21 f is mono if and only if f is injective on underlying sets. ∎

Proposition 4.3.22 *f is epi if and only if f is surjective on underlying sets.*

Proof. If the underlying function $f\colon X \to Y$ is surjective, it is almost immediate that f is epi. On the other hand, suppose f is epi in ω-*Sets*, let Z be a set and let

$$g, h\colon Y \rightrightarrows Z$$

be set functions such that $gf = gh$. These lift to arrows

$$g, h\colon (Y, \in) \rightrightarrows \nabla Z$$

in ω-*Sets*, and $gf = gh$ in ω-*Sets*; so $g = h$ as arrows in ω-*Sets*, and hence as functions between the underlying sets. Hence f is a surjective function. ∎

Example 4.3.23 Let $A \subseteq \mathbb{N}$ and define an ω-set $(X, \in)$ by letting $X = \mathbb{N}$ and putting

$$|n \in X| = \begin{cases} \{\langle 0, n\rangle\} & n \notin A \\ \{\langle 1, n\rangle\} & n \in A \end{cases}$$

There is an obvious arrow $i\colon (X, \in) \to N$ tracked by (a code for) the second projection $\langle m, n\rangle \mapsto n$. As we have just seen, this arrow is both mono and epi. However, suppose it had an inverse $j\colon N \to (X, \in)$, tracked by a code $e \in \mathbb{N}$. Then

$$\{e\}(n) = \begin{cases} \langle 0, n\rangle & n \notin A \\ \langle 1, n\rangle & n \in A \end{cases}$$

If A is not recursive (e.g. if A is the halting set) then no such code e can exist. ∎

Exercise 4.3.24 Give an example of a partial map $(X, \in) \rightharpoonup (Y, \in)$ in ω-*Sets* that is *not* the restriction of a (total) arrow $(X, \in) \to (Y, \in)$ to a subobject of $(X, \in)$. (This example will carry over to $\mathcal{E}\!f\!f$.) ∎

4.4 The effective topos

So ω-$\mathcal{S}ets$ won't do. One solution is to enlarge it, by freely adjoining 'good' quotients of equivalence relations, to obtain a topos: [4]. Another is to construct a topos from the ground up, as Hyland did, and then show that it contains ω-$\mathcal{S}ets$ as a full subcategory. Either way turns out to give the same answer. We will take the second route, as it gives us a better characterisation of the internal logic of the resulting topos, and is thus more convenient for logical calculations.

As we'll see, general objects of $\mathcal{E}\!f\!f$ are rather cumbersome to work with. Fortunately, nearly all the calculations we need to perform in practice involve only ω-sets, for which substantial simplifications are possible. We'll see this as we go along.

The full details of the construction of a general realizability topos will be dealt with in separate notes. Some of the proofs are quite technical. What follows should be taken as an outline, and an explanation of the basic ideas. Most of the results appear in [7], though not the embedding of ω-$\mathcal{S}ets$.

We begin by redescribing the propositional part of realizability in more familiar terms, using an idea of Dana Scott's. We'll need a slight generalisation of the notion of Heyting algebra.

Definition 4.4.1 *A **Heyting prealgebra** H is a preorder with finite products $(\wedge, \top)$, finite coproducts $(\vee, \bot)$ and exponentials $(\rightarrow)$. If $p, q \in H$ we write*

$$p \equiv q$$

to indicate that $p \leq q$ and $q \leq p$. Note that $p \equiv q$ iff $p \leftrightarrow q \equiv \top$.

It's easy to check that any Heyting prealgebra H gives us an interpretation of constructive propositional logic, as before. So if ϕ is a propositional formula, we know what it means to say that

$$H \models \phi$$

The particular Heyting prealgebra we're interested in is one whose definition is analogous to our earlier definition of recursive realizability.

Definition 4.4.2 *We can turn $\boldsymbol{P}\mathbb{N}$ into a Heyting prealgebra as follows: for $p, q \subseteq \mathbb{N}$, define*

$$\begin{array}{rcl} p \wedge q & = & \{\langle m, n\rangle \mid m \in p, n \in q\} \\ p \vee q & = & \{\langle 0, m\rangle \mid m \in p\} \cup \{\langle 1, n\rangle \mid n \in q\} \\ p \rightarrow q & = & \{e \mid \text{for all } n \in p,\ \{e\}(n) \in q\} \end{array}$$

and let

$$p \leq q \quad \text{iff} \quad p \rightarrow q \neq \varnothing$$

Clearly $\bot = \varnothing$ is a bottom element and $\top = \mathbb{N}$ is a top element (as is any other nonempty subset; $\boldsymbol{P}\mathbb{N}$ is actually equivalent to the 2-element poset).

This definition is a little misleading. In order to interpret quantification, we really need a Heyting prealgebra structure on $(\mathbf{P}\mathbb{N})^X$, for each set X. We could take the pointwise structure, but that turns out to be the wrong one! The correct definition is the following.

Definition 4.4.3 *If X is a set, we turn $(\boldsymbol{P}\mathbb{N})^X$ into a Heyting prealgebra in the following way: if $\phi, \psi \in (\boldsymbol{P}\mathbb{N})^X$, so that $F, G \subseteq \mathbb{N}$ for each $x \in X$, we define*

$$\begin{array}{rcl} (F \wedge G)(x) & = & F(x) \wedge G(x) \\ (F \vee G)(x) & = & F(x) \vee G(x) \\ (F \rightarrow G)(x) & = & F(x) \rightarrow G(x) \end{array}$$

and we let

$$F \leq G \quad \textit{iff} \quad \bigcap_{x \in X} F(x) \rightarrow G(x) \neq \varnothing$$

Clearly $\perp = \lambda x.\varnothing \in (\boldsymbol{P}\mathbb{N})^X$ is a bottom element, and $\top = \lambda x.\mathbb{N}$ is a top element.

We will also abbreviate

$$\mathbf{P}\mathbb{N} \models (F \rightarrow G) \wedge (G \rightarrow F)$$

by $\mathbf{P}\mathbb{N} \models F \leftrightarrow G$.

Note that the definition of the preorder $\leq$ is 'uniform' rather than pointwise: a *single* code must work for all $x \in X$. This is what makes the structure of $(\mathbf{P}\mathbb{N})^X$ interesting. In fact, even the structure of $(\mathbf{P}\mathbb{N})^2$ is quite mysterious.

Anyway, this corresponds (though not precisely) to our previous definition of realizability for open formulas: an open formula is realizable iff there is a 'uniform' realizer for each of its instances. It's convenient, and hopefully not confusing, to make the following definition.

Definition 4.4.4 *If $F \in (\boldsymbol{P}\mathbb{N})^X$, we say that*

$$\boldsymbol{P}\mathbb{N} \models F$$

if $\bigcap_{x \in X} F(x) \neq \varnothing$. If $F = \lambda x.F(x)$ we often just write $\boldsymbol{P}\mathbb{N} \models F(x)$.

Definition 4.4.5 *If $\boldsymbol{P}\mathbb{N} \models F$, a* ***realizer*** *for F is an element $e \in \bigcap_{x \in X} F(x)$. We also say that e* ***realizes*** *F.*

Exercise 4.4.6 If $F \in \mathbf{P}\mathbb{N}^X$, explain why $\mathbf{P}\mathbb{N} \models F$ iff $\top \leq F$. ■

Definition of the effective topos

Definition 4.4.7 *An* ***object*** *$(X, =)$ of $\mathcal{E}\!f\!f$ is a set X together with a* ***($\boldsymbol{P}\mathbb{N}$-valued) equality relation****, i.e. for each $x, x' \in X$ a subset*

$$|x = x'| \subseteq \mathbb{N}$$

that is "realizably symmetric and transitive": this means that

$$\boldsymbol{P}\mathbb{N} \models |x = x'| \rightarrow |x' = x|$$

and

$$\boldsymbol{P}\mathbb{N} \models (|x = x'| \wedge |x' = x''|) \rightarrow |x = x''|$$

If $x \in X$, we write Ex for $|x = x|$.

Recall what all this means: there is a *single* code $e \in \mathbb{N}$ that, for all $x, x' \in X$, maps $|x = x'|$ into $|x' = x|$, and so on. The relation $=$ is not required to be 'reflexive', so Ex turns out to be a nontrivial 'existence predicate', cf. the Fourman/Scott theory of identity and existence. We could also write "$|x \in X|$" for Ex (this will turn out to be consistent with our earlier notation for ω-sets) but we won't.

(Actually, for technical reasons it is better to let an object of $\mathcal{E}\!f\!f$ be a set endowed with an $\equiv$-*equivalence class* of equality relations, cf. the following definition. However, for our calculations it's easier to stick with the above definition. The categories we get in the end are of course equivalent.)

Definition 4.4.8 *Suppose $(X,=),(Y,=) \in \mathcal{E}\!f\!f$. Then a **($\boldsymbol{P}\mathbb{N}$-valued) functional relation** from $(X,=)$ to $(Y,=)$ is a map $F\colon X \times Y \to \boldsymbol{P}\mathbb{N}$ such that*

*F is **relational:***	$\boldsymbol{P}\mathbb{N} \models (F(x,y) \wedge (|x = x'| \wedge |y = y'|)) \to F(x',y')$
*F is **strict:***	$\boldsymbol{P}\mathbb{N} \models F(x,y) \to (Ex \wedge Ey)$
*F is **single-valued:***	$\boldsymbol{P}\mathbb{N} \models F(x,y) \wedge F(x,y') \to |y = y'|$
*F is **total:***	$\boldsymbol{P}\mathbb{N} \models Ex \to \{\langle k,m\rangle \mid k \in Ey \text{ for some } y \in Y, \text{ and } m \in F(x,y)\}$

*Two functional relations F, F' are **equivalent**, written $F \sim F'$, if*

$$\boldsymbol{P}\mathbb{N} \models (F(x,y) \leftrightarrow F'(x,y)$$

Note that we could have defined 'total' to mean

$$\mathbf{P}\mathbb{N} \models Ex \to \bigcup_{y \in Y} (Ey \wedge F(x,y))$$

(cf. our characterisation of the logic of $\mathcal{E}\!f\!f$, in the next section.)

Definition 4.4.9 *The **identity** functional relation I from $(X,=)$ to $(X,=)$ is defined by*

$$I(x,x') = |x = x'|$$

Proposition 4.4.10 *If F is a functional relation from $(X,=)$ to $(Y,=)$, and G is a functional relation from $(Y,=)$ to $(Z,=)$, then we can define a **composite** functional relation $G \circ F$ from $(X,=)$ to $(Z,=)$. Moreover, if $F \sim F'$ and $G \sim G'$ then $G \circ F \sim G' \circ F'$.*

Proof. (Sketch.) We let

$$\begin{aligned}(G \circ F)(x,z) &= \text{“}\exists y.\, F(x,y) \wedge G(y,z)\text{”} \\ &= \{\langle k,m,n\rangle \mid k \in |y = y| \text{ for some } y \in Y, \text{ and } m \in F(x,y), n \in G(y,z)\}\end{aligned}$$

■

So the following definition gives us a category $\mathcal{E}\!f\!f$.

Definition 4.4.11 *An **arrow** $f\colon (X,=) \to (Y,=)$ in $\mathcal{E}\!f\!f$ is an equivalence class of functional relations. We often write $f = [F]$ where F is a functional relation representing f.*

The following observation often saves us some work.

Exercise 4.4.12 Let $[F], [G]: (X, =) \to (Y, =) \in \mathcal{E}\!f\!f$. Then $F \sim G$ iff

$$\mathbf{P}\mathbb{N} \models F(x, y) \to G(x, y)$$

(Hint: G is strict and single-valued and F is total.) ∎

Note that we *cannot* think of an arrow $(X, =) \to (Y, =)$ as coming from a function $f: X \to Y$ between the underlying sets anymore. This is the main reason why general objects of $\mathcal{E}\!f\!f$ are much more unwieldy to calculate with than ω-sets.

The main theorem—which is pretty hard—is the following:

Theorem 4.4.13 (Hyland) *(See [9]) $\mathcal{E}\!f\!f$ is a topos.*

Some objects in the effective topos

Proposition 4.4.14 *Define $1 = (\{*\}, =) \in \mathcal{E}\!f\!f$ by letting*

$$|* = *| = \mathbb{N}$$

Then 1 is a terminal object in $\mathcal{E}\!f\!f$.

Proof. Given $(X, =) \in \mathcal{E}\!f\!f$, define $!: X \times \{*\} \to \mathbf{P}\mathbb{N}$ by

$$!(x, *) = |x = x|$$

One checks that $[!]: (X, =) \to 1$ and that it is the unique arrow. ∎

Proposition 4.4.15 *Let $(X, =), (Y, =) \in \mathcal{E}\!f\!f$. Then their product can be defined to be $(X \times Y, =)$, where*

$$|\langle x, y\rangle = \langle x', y'\rangle| = |x = x'| \wedge |y = y'|$$

The proof, as well as the construction of finite limits, exponentials etc., is omitted.

Proposition 4.4.16 *Let $\Omega = (\boldsymbol{P}\mathbb{N}, =)$, where if $p, q \in \boldsymbol{P}\mathbb{N}$,*

$$|p = q| = p \leftrightarrow q$$

Then Ω is a subobject classifier in $\mathcal{E}\!f\!f$.

The natural numbers object Let $N = (\mathbb{N}, =) \in \mathcal{E}\!f\!f$ where

$$|n = n'| = \begin{cases} \{n\} & n = n' \\ \varnothing & n \neq n' \end{cases}$$

(This is actually a special case of a construction we are about to describe soon.) Recall definition 4.3.3, where we described the natural numbers object N in ω-*Sets*. We're about to see how the ω-set N gives rise to an object of $\mathcal{E}\!f\!f$, which turns out to be the object N we just defined. But first, the theorem.

Proposition 4.4.17 *Define arrows $[O]: 1 \to N$, $[S]: N \to N$ in $\mathcal{E}\!f\!f$ as follows:*

$$O(*, n) = \begin{cases} \{0\} & n = 0 \\ \varnothing & n \neq 0 \end{cases}$$
$$S(n, m) = \begin{cases} \{n+1\} & m = n+1 \\ \varnothing & m \neq n+1 \end{cases}$$

With these arrows, N is a natural numbers object in $\mathcal{E}\!f\!f$.

Proof. Suppose $(X, =) \in \mathcal{E}\!f\!f$ and we have $[F]: 1 \to (X, =)$, $[G]: (X, =) \to (X, =)$; we want to define $[H]: N \to (X, =)$. First note that the n-fold composite $[G]^n: (X, =) \to (X, =)$ is $[G_n]$ where

$$G_0(x, x') = |x = x'|$$
$$G_n(x, x') = \{\langle k, l\rangle \mid k \in Ex'', l \in G_{n-1}(x, x'') \wedge G(x'', x)\}$$

We define

$$H(n, x) = \{n\} \wedge \{\langle k, l\rangle \mid k \in Ex' \text{ for some } x' \in X, \text{ and } l \in F(*) \wedge G_n(x', x)\}$$

The proofs that H a functional relation, that $[H]$ makes the appropriate diagram commute and that $[H]$ is the unique such arrow would, with our present tools, be tedious. We can give a fairly simple version once we have the notion of a realizability tripos, so we refer the reader to those nonexistent notes for the rest of the proof. ■

Although this result does not follow from proposition 4.3.4, $[O]$ and $[S]$ actually come from the maps $0: 1 \to N$, $s: N \to N \in \omega\text{-}\mathcal{S}ets$ defined in 4.3.4, via the following construction:

ω-sets as objects in $\mathcal{E}\!f\!f$

We can define a full embedding

$$\omega\text{-}\mathcal{S}ets \hookrightarrow \mathcal{E}\!f\!f$$

as follows (cf. the definition of N): $(X, \in) \mapsto (X, =)$ where

$$|x = x'| = \begin{cases} |x \in X| & x = x' \\ \varnothing & x \neq x' \end{cases}$$

and if e tracks a map $f: (X, \in) \to (Y, \in)$, we put

$$F(x, y) = \begin{cases} \{\langle n, \{e\}(n)\rangle \mid n \in |x \in X|\} & f(x) = y \\ \varnothing & f(x) \neq y \end{cases}$$

F 'encodes the graph of f'. We need to check the following:

Lemma 4.4.18 *F, as defined, is a functional relation from $(X, =)$ to $(Y, =)$.*

Proof. We need to check that F is relational, strict, single-valued and total, as in definition 4.4.8. We will construct a realizer for each statement in turn.

By the construction of $(X, =)$ and $(Y, =)$,

$$|x = x'| \neq \varnothing, |y = y'| \neq \varnothing$$

implies $x = x', y = y'$, and hence $F(x, y) = F(x', y')$. So $\Lambda n.n_0$ realizes

$$(F(x,y) \wedge (|x = x'| \wedge |y = y'|)) \rightarrow F(x', y')$$

Observe that if $x \in X$, $Ex = |x = x| = |x \in X|$. So by the fact that e tracks f, and the definition of F, $\Lambda n.n$ realizes

$$F(x, y) \rightarrow (Ex \wedge Ey)$$

If $F(x, y) \wedge F(x, y') \neq \varnothing$ then $y = f(x) = y'$, so $\Lambda n.(n_0)_1$ realizes

$$F(x, y) \wedge F(x, y') \rightarrow |y = y'|$$

Finally, $\Lambda n.\langle n, \{e\}(n)\rangle$ realizes

$$Ex \rightarrow \{\langle k, m\rangle |\, k \in Ey \text{ for some } y \in Y, \text{ and } m \in F(x, y)\}$$

■

Lemma 4.4.19 *The embedding is full.*

Proof. Let F be a functional relation from $(X, =)$ to $(Y, =)$, which come from ω-sets as described above. We want to define an arrow $f\colon (X, \in) \rightarrow (Y, \in)$.

F is total. So let e be a realizer for

$$Ex \rightarrow \{\langle k, m\rangle : k \in Ey \text{ for some } y \in Y, \text{ and } m \in F(x, y)\}$$

We can define a function $f\colon X \rightarrow Y$ as follows: given $x \in X$, pick some $n \in |x \in X|$ and let $f(x)$ be some $y \in Y$ such that $\{e\}(n) = \langle k, m\rangle$ with $k \in Ey = |y \in Y|$. This actually gives us an arrow

$$f\colon (X, \in) \rightarrow (Y, \in)$$

tracked by $\Lambda n.(\{e\}(n))_0$. We want to show that f is the only function we could have defined, i.e. that the choice of y was unique.

So let e' be another realizer for

$$Ex \rightarrow \{\langle k, m\rangle : k \in Ey \text{ for some } y \in Y, \text{ and } m \in F(x, y)\}$$

and choose, as before, some $n' \in |x \in X|$ and some $y' \in Y$ such that $\{e'\}(n') = \langle k', m'\rangle$ with $k' \in |y' \in Y|$. Then $m \in F(x, y)$ and $m' \in F(x, y')$.

F is single-valued. So let r be a realizer for

$$F(x, y) \wedge F(x, y') \rightarrow |y = y'|$$

Then $\{r\}(\langle m, m'\rangle) \in |y = y'|$, i.e. $|y = y'| \neq \varnothing$. So $y = y'$. ■

It will turn out that ω-$\mathcal{S}ets$ is a full reflective subcategory of $\mathcal{E}\!f\!f$, and that the inclusion preserves exponentials (and local exponentials), again for general reasons: see section 4.5.

4.5 More on the effective topos

Truth in the effective topos

We will now give a concrete description of what

$$\mathcal{E}\!f\!f \models \phi$$

means; that is, of what it means for a code $n \in \mathbb{N}$ to 'realize' a formula in the internal language of $\mathcal{E}\!f\!f$. In doing this, we will see exactly how the logic of $\mathcal{E}\!f\!f$ extends our earlier realizability interpretation for first-order arithmetic.

We begin with a characterisation of predicates in $\mathcal{E}\!f\!f$. The proof, which is messy, is omitted. Notice that the following definition generalises the definition of functional relation in 4.4.8.

Definition 4.5.1 *Let $(X, =) \in \mathcal{E}\!f\!f$. A **strict relation** on $(X, =)$ is a function $R: X \to \boldsymbol{P}\mathbb{N}$ that respects equality:*

$$\boldsymbol{P}\mathbb{N} \models (R(x) \wedge |x = x'|) \to R(x')$$

and existence:

$$\boldsymbol{P}\mathbb{N} \models R(x) \to Ex$$

*Two strict relations R, R' are **equivalent** if*

$$\boldsymbol{P}\mathbb{N} \models R(x) \leftrightarrow R'(x)$$

Example 4.5.2 A strict relation on an ω-set $(X, \in)$ is just a function $R: X \to \mathbf{P}\mathbb{N}$. ∎

Proposition 4.5.3 *(See [7, proposition 14.2]) Predicates $\phi: X \to \Omega$ are in bijective correspondence with equivalence classes $[R]$ of strict relations on $(X, =)$.*

This means that it suffices to deal with strict relations on $(X, =)$, which are more concrete things and easier to work with than maps into Ω. If $\phi, \psi: (X, =) \to \Omega$ we will write R, S for corresponding strict relations. It does not matter which representatives we choose.

Of course, we still need to know *which* strict relations represent to a given predicate:

Example 4.5.4 If $(X, =) \in \mathcal{E}\!f\!f$, the equality predicate $(X, =) \times (X, =) \to \Omega$ is represented by the strict relation $X \times X \to \mathbf{P}\mathbb{N}$ taking

$$x, x' \mapsto |x = x'|$$

And if $f = [F]: (X, =) \to (Y, =)$, the predicate $(X, =) \times (Y, =) \to \Omega$ classifying the graph of f (i.e. the predicate "$f(x) = y$") is represented by the strict relation

$$x, y \mapsto F(x, y)$$

on $(X, =) \times (Y, =)$. These are all the atomic predicates we'll need to use. Note that since we have no analogue of 4.5.3 for arbitrary maps in $\mathcal{E}\!f\!f$ (no way of knowing they come from functions $X \to Y$) we had to treat "$f(x) = y$" separately from "$x = x$". ∎

Example 4.5.5 If $(X, \in)$ is an ω-set, regarded as an object of $\mathcal{E}\!f\!f$, then the equality predicate on $(X, \in)$ is represented by the function $E: X \times X \to \mathbf{P}\mathbb{N}$ taking

$$E(x, x') = \begin{cases} |x \in X| & x = x' \\ \varnothing & x \neq x' \end{cases}$$

If $f: (X, \in) \to (Y, \in)$ is an arrow between ω-sets, the predicate "$f(x) = y$" is represented by the function $F: X \times Y \to \mathbf{P}\mathbb{N}$ taking

$$F(x, y) = \begin{cases} \{\langle n, \{e\}(n)\rangle : n \in |x \in X|\} & f(x) = y \\ \varnothing & f(x) \neq y \end{cases}$$

where $e \in \mathbb{N}$ tracks f. This is immediate from the embedding $\omega\text{-}\mathcal{S}ets \hookrightarrow \mathcal{E}\!f\!f$. ■

The strict relations corresponding to compound (propositional) formulas are constructed using the Heyting prealgebra operations in $\mathbf{P}\mathbb{N}^X$, in the obvious way:

Proposition 4.5.6 *Let ϕ, ψ be predicates and R, S corresponding strict relations. Then*

$$\begin{array}{rcl} \phi \wedge \psi & \textit{corresp.} & R \wedge S \\ \phi \vee \psi & \textit{corresp.} & R \vee S \\ \phi \to \psi & \textit{corresp.} & R \to S \end{array}$$

For the predicate $\bot$, we can just take the strict relation $\lambda x. \varnothing$.

Exercise 4.5.7 What about andalso? ■

It remains to interpret the quantifiers:

Proposition 4.5.8 *Suppose the predicate $\phi: (X, =) \times (Y, =) \to \Omega$ is represented by the strict relation $R: X \times Y \to \boldsymbol{P}\mathbb{N}$. Then*

$$\begin{array}{lcl} \forall x : (X, =). \, \phi(x, y) & \textit{corresp.} & \bigcap_{x \in X} Ex \to R(x, y) \\ \exists x : (X, =). \, \phi(x, y) & \textit{corresp.} & \bigcup_{x \in X} Ex \wedge R(x, y) \end{array}$$

(Remember that these expressions define functions $Y \to \boldsymbol{P}\mathbb{N}$.)

Finally, the characterisation of validity:

Proposition 4.5.9 *Let ϕ be a predicate on $(X, =) \in \mathcal{E}\!f\!f$, corresponding to a strict relation $R \in \boldsymbol{P}\mathbb{N}^X$. Then $\mathcal{E}\!f\!f \models \phi$ iff $\boldsymbol{P}\mathbb{N} \models R$, i.e. iff*

$$\bigcap_{x \in X} Ex \to R(x) \neq \varnothing$$

An element of this set is called a ***realizer*** *for R.*

(Note that if ϕ is a predicate on $(X, =) \times (Y, =)$, $\mathcal{E}\!f\!f \models \phi$ iff

$$\bigcap_{x \in X, y \in Y} Ex \wedge Ey \to R(x, y)$$

and so on. This follows easily from the definition of products in $\mathcal{E}\!f\!f$.)

So in order to show that $\mathcal{E}\!f\!f \models \phi$, we simply have to find a suitable realizer for a corresponding strict relation R, constructed in accordance with the recipe just prescribed. We will see some complicated examples a bit later, when we go back to the principles of recursive mathematics to see whether they're valid in $\mathcal{E}\!f\!f$. First, some simpler examples of the characterisation in action.

Exercise 4.5.10 Let $(X, =) \in \mathcal{E}\!f\!f$ come from an ω-set $(X, \in)$. Show that the predicate "$(x = x') \vee (x \neq x')$" is represented by the function $D: X \times X \to \mathbf{P}\mathbb{N}$ taking

$$D(x, x') = \begin{cases} \{\langle 0, n\rangle : n \in |x \in X|\} & x = x' \\ \{\langle 1, e\rangle : e \in \mathbb{N}\} & x \neq x' \end{cases}$$

Deduce that

$$\mathcal{E}\!f\!f \models \forall x, x' : (X, =). (x = x') \vee (x \neq x')$$

iff $(X, \in)$ is decidable (see exercise 4.3.8). ∎

Exercise 4.5.11 Recall the system of first-order arithmetic described in section 4.1. For each formula ϕ in that language, define a corresponding formula ϕ^* in the internal language of $\mathcal{E}\!f\!f$. By using the description of N and the characterisation of validity in $\mathcal{E}\!f\!f$, prove that ϕ is realizable iff $\mathcal{E}\!f\!f \models \phi^*$. This shows that the logic of $\mathcal{E}\!f\!f$ really does extend the realizability interpretation of first-order arithmetic. ∎

Monos, epis, subobjects and quotients

Example 4.5.12 Let $f: (X, =) \to (Y, =)$ be an arrow in $\mathcal{E}\!f\!f$. Recall that f is mono iff

$$\mathcal{E}\!f\!f \models \forall x, x' : (X, =). f(x) = f(x') \to x = x'$$

which we can rewrite as

$$\mathcal{E}\!f\!f \models \forall x, x' : (X, =). \forall y : (Y, =). (f(x) = y \wedge f(x') = y) \to x = x'$$

Suppose that $f = [F]$, where F is a functional relation from $(X, =)$ to $(Y, =)$. Unwinding the characterisation of validity in $\mathcal{E}\!f\!f$, this holds iff there is an element of

$$\bigcap_{x, x' \in X} \bigcap_{y \in Y} Ex \wedge Ex' \wedge Ey \wedge F(x, y) \wedge F(x', y) \to |x = x'|$$

(we have done a little rearranging and uncurrying). This is more complicated than we might hope; luckily any mono is always isomorphic to one which can be defined in a much simpler way. Suppose that f is classified by $\phi: (Y, =) \to \Omega$, where (by 4.5.3) ϕ is represented by a strict relation $R: Y \to \mathbf{P}\mathbb{N}$. Then $(X, =) \cong \|\phi\|$, and $\|\phi\|$ may be described explicitly as $(Y, =_R)$ where

$$|y =_R y'| = R(y) \wedge |y = y'|$$

∎

Example 4.5.13 If $(X, \in)$ is an ω-set and we have a predicate ϕ on $(X, \in)$ represented by a function $R: X \to \mathbf{P}\mathbb{N}$, then we can let $\|\phi\| = (X', \in')$ where

$$X' = \{x \in X \,|\, R(x) \neq \varnothing\}$$

and

$$|x \in' X'| = R(x) \wedge |x \in X|$$

(The only reason for restricting ourselves to $X' \subseteq X$ was to make sure that this set is nonempty. Also, note that even if $X' = X$, we do not necessarily get something isomorphic to $(X, \in)$; can you think of an example?) ∎

Example 4.5.14 Now recall that f is epi iff

$$\mathcal{E}\!f\!f \models \forall y : (Y, =).\, \exists x : (X, =).\, f(x) = y$$

i.e. if we have an element of

$$\bigcap_{y \in Y} Ey \to \Big(\bigcup_{x \in X} Ex \wedge F(x, y)\Big)$$

This is also a little more complicated than we might hope; luckily any epi is also always isomorphic to one which can be defined in a much simpler way. Recall that every epi in a topos is the coequalizer of its kernel pair, i.e. the quotient of an equivalence relation. That is, an epi $f\colon (X, =) \to (Y, =)$ corresponds to an equivalence relation $\approx\colon (X, =) \times (X, =) \to \Omega$, i.e. to a strict relation $\approx\colon X \times X \to \mathbf{P}\mathbb{N}$ such that

$$\begin{gathered} \mathbf{P}\mathbb{N} \models Ex \to |x \approx x'| \\ \mathbf{P}\mathbb{N} \models Ex \wedge Ex' \wedge |x \approx x'| \to |x' \approx x| \\ \mathbf{P}\mathbb{N} \models Ex \wedge Ex' \wedge Ex'' \wedge |x \approx x'| \wedge |x' \approx x''| \to |x \approx x''| \end{gathered}$$

Then $(X, \approx) \in \mathcal{E}\!f\!f$, $\approx$ is a functional relation from $(X, =)$ to $(X, \approx)$:

$$[\approx]\colon (X, =) \to (X, \approx)$$

and $(Y, =) \cong (X, \approx)$. ■

In the following example (and quite frequently thereafter) we confuse an ω-set with the object of $\mathcal{E}\!f\!f$ to which it gives rise. This simplifies the notation.

Example 4.5.15 It follows that any object $(X, =) \in \mathcal{E}\!f\!f$ is a quotient of a subobject of an 'ordinary set', in the following sense. Define an ω-set $(X, \in)$ by putting

$$|x \in X| = Ex$$

We then have a mono into the 'ordinary set' ∇X,

$$(X, \in) \rightarrowtail \nabla X$$

(tracked by, e.g., $\Lambda n.n$) and an epi

$$[=]\colon (X, \in) \twoheadrightarrow (X, =)$$

$(X, \in)$ is called the **separated cover** of $(X, =)$. Notice that it is *not* canonical, as it depends on the choice of a particular representation for $(X, =)$. ■

Revisiting some constructive principles

We'll try to justify the claim that "mathematics in $\mathcal{E}\!f\!f$ is recursive mathematics" (which, it seems from [7], was one of the main motivations for constructing $\mathcal{E}\!f\!f$) by proving the validity of some principles used in recursive mathematics. We'll start with Church's Thesis. We could verify the first-order version, as we did earlier, but for interest we look at the higher-order version.

Proposition 4.5.16 *In $\mathcal{E}\!f\!f$, Church's Thesis is valid in the following form:*

$$\forall f : N^N.\, \exists e : N.\, \forall n : N.\, \exists y : N.\, T(e, n, y) \wedge f(n) = U(y)$$

Proof. First some preliminaries. Primitive recursive functions can be represented in $\mathcal{E}\!f\!f$, so the T-predicate and U-function are definable. In fact, since $T(e,n,y) \leftrightarrow (t(e,n,y)=0)$ for a suitable primitive recursive function t, $T(e,n,y)$ is decidable and hence $\neg\neg$-closed; so[8] we may assume that it comes from the following strict relation $T\colon \mathbb{N}\times\mathbb{N}\times\mathbb{N}\to \mathbf{P}\mathbb{N}$:

$$T(e,n,y) = \begin{cases} \{0\} & \text{if } T(e,n,y) \\ \varnothing & \text{otherwise} \end{cases}$$

(We implicitly assume that T and U "mean the same in $\mathcal{E}\!f\!f$ as they do in $\mathcal{S}ets$"; this is intuitively clear, but a rigorous statement and proof requires tools not developed here.) Also, let $e' = \Lambda e, n.\, (\mu y. T(e,n,y))$.

Next, since the inclusion $\mathsf{Per} \hookrightarrow \mathcal{E}\!f\!f$ preserves exponentials, $N^N = (T, =)$ where T is the set of total recursive functions $f\colon \mathbb{N}\to\mathbb{N}$ and

$$|f = f'| = \begin{cases} \{e \mid e \text{ is a code for } f\} & f = f' \\ \varnothing & f \neq f' \end{cases}$$

Now to work. We need an element of

$$\bigcap_{f\in T} \Big(Ef \to \bigcup_e \Big(\{e\} \wedge \bigcap_n (\{n\} \to \bigcup_y \{y\} \wedge (T(e,n,y) \wedge |f(n) = U(y)|))\Big)\Big)$$

Suppose $f \in T$ and $e \in Ef$ (i.e. e is a code for f). Then for all $n \in \mathbb{N}$, $y = \{e'\}(e,n)$ is defined. So we can define our realizer to be

$$\Lambda e. \langle e, \Lambda n. \langle \{e'\}(e,n), \langle 0, \{e\}(n)\rangle\rangle\rangle$$

(Note the slight messiness involved.) ∎

Extended Church's Thesis is also valid (both first-order and higher-order versions): see [7, lemma 10.3]. We won't spell it out. There is a related result, valid in any realizability topos, which turns out to be more useful; see lemma 4.6.7.

Principles whose validity will be proved elsewhere We'll list version of some more useful principles which are valid in $\mathcal{E}\!f\!f$. The proofs are postponed to other notes because

- they require more information about the structure of $\mathcal{E}\!f\!f$ than we have, and
- unlike Church's Thesis, they are valid in any realizability topos.

First we have **countable choice**:

$$\mathcal{E}\!f\!f \models (\forall n : N.\, \exists x : (X,=).\, \phi(n,x)) \to \exists f : (X,=)^N.\, \forall n : N.\, \phi(n, f(n))$$

and the more general **dependent choices** (which makes most of analysis possible):

$$\begin{aligned} \mathcal{E}\!f\!f \models\ & (\phi(x_0) \wedge \forall x : (X,=).\, (\phi(x) \to \exists x' : (X,=).\, \psi(x,x') \wedge \phi(x'))) \\ & \to \exists f : (X,=)^N.\, f(0) = x_0 \wedge \forall n : N.\, \psi(f(n), f(n+1)) \end{aligned}$$

[8] strictly speaking, by proposition 4.5.25

Next is **Markov's Principle**, which we saw in proposition 4.1.13:

$$\mathcal{E}\!f\!f \models \forall r : PN.\, ((\forall n : N.\, n \in r \vee n \notin r) \wedge \neg\neg(\exists n : N.\, n \in r) \rightarrow \exists n : N.\, n \in r)$$

Finally we have Troelstra's **uniformity principle** (cf. exercise 4.3.12):

$$\mathcal{E}\!f\!f \models (\forall r : PN.\, \exists n : N.\, \phi(n, r)) \rightarrow \exists n : N.\, \forall r : PN.\, \phi(n, r)$$

This looks odd at first. The idea is that subsets of N are not very easy to characterise—there are any number of ways to carve out a subset of N. On the other hand, the natural numbers are perfectly definite, and even form a decidable set. So if we must assign a number to *every* subset, in an effective way, the only way to do this is to use the same number for each one.

Example 4.5.17 Let's see why the following variant of the uniformity principle is valid:

$$\forall x : \nabla X.\, \exists n : N.\, \phi(x, n) \rightarrow \exists n : N.\, \forall x : \nabla X.\, \phi(x, n)$$

Suppose ϕ comes from a strict relation $R\colon X \times \mathbb{N} \rightarrow \mathbf{P}\mathbb{N}$, and suppose we have

$$e \in \bigcap_{x \in X} Ex \rightarrow \bigcup_{n \in \mathbb{N}} En \wedge R(x, n)$$

(where $Ex = \mathbb{N}$ and $En = \{n\}$). We need an effective way of computing an element

$$k \in \bigcup_{n \in \mathbb{N}} En \wedge \bigcap_{x \in X} Ex \rightarrow R(x, n)$$

Now, since $0 \in Ex$ for all $x \in X$, $\{e\}(0) = \langle n_0, m \rangle$ where $m \in R(x, n_0)$ for all x. So we can put

$$k = \langle n_0, \Lambda y.\, m \rangle$$

(Note: all we needed to know about ∇X was the fact that $\bigcap_{x \in X} Ex \neq \emptyset$.) ∎

Exercise 4.5.18 If $n : N, r : PN$ let $\phi(n, r)$ be the predicate

$$(n = 0 \wedge r = \emptyset) \vee (n = 1 \wedge r \neq \emptyset)$$

Why doesn't this give us a counterexample to the uniformity principle? ∎

Example 4.5.19 Let's see why countable choice is valid in the case $(X, =) = N$. Suppose ϕ has no extra free variables, ϕ corresponds to the strict relation $F\colon \mathbb{N} \times \mathbb{N} \rightarrow \mathbf{P}\mathbb{N}$ and e realizes

$$\bigcap_n \{n\} \rightarrow (\bigcup_m \{m\} \wedge F(n, m))$$

This means that for each $n \in \mathbb{N}$, $\{e\}(n) = \langle m, k \rangle$ where $k \in F(n, m)$.

Recall the description of N^N from the proof of 4.5.16. We must find a realizer for

$$\bigcap_{f \in T} Ef \wedge (\bigcap_n \{n\} \rightarrow F(n, f(n)))$$

Clearly we can put $f = \{e\}$ and let the realizer be $\langle e, e \rangle$. So $\Lambda e.\langle e, e \rangle$ realizes countable choice. (For an arbitrary $(X, =)$ our task is harder, since $(X, =)^N$ is a lot more complicated[9].) ∎

[9]Interestingly, the general proof requires the countable axiom of choice in the *metatheory*: see [7, section 9]

Let us mention some of the consequences, without giving too many details (see [7, section 9]). In $\mathcal{E}\!f\!f$, we can define the integers, the rationals and the real numbers (equivalence classes of Cauchy sequences of rationals); we can then go on to develop complex analysis and prove, for example, the fundamental theorem of algebra—entirely within the internal language of a topos, using the additional principles just mentioned. In fact, only dependent choices is needed: Church's Thesis and Markov's Principle are unnecessary.

By unwinding the definitions and the characterisation of validity in $\mathcal{E}\!f\!f$, we can then verify that the real numbers in $\mathcal{E}\!f\!f$ correspond to *recursive* reals, and the "fundamental theorem of algebra in $\mathcal{E}\!f\!f$" translates to the statement that a polynomial over $\mathbb{C}$ with recursive coefficients has a recursive root. Moreover, from the constructive proof, we can extract an index for a recursive function that *computes* such a root.

Notice the advantage of taking this approach towards recursive mathematics: we don't need to work with enumerations, or mention recursive functions explicitly—we simply note that the ordinary constructive proofs may be formalised in the internal language of $\mathcal{E}\!f\!f$. This not only saves a lot of work but gives us a precise way of stating the relationship between recursive mathematics and (Bishop-style) constructive mathematics. It also helps us formulate concepts in the correct way: to choose, out of many possibilities, the 'right' definition of a concept in the context of recursive mathematics, we may take the corresponding concept in constructive mathematics (where 'right' choice is hopefully already known) and use the logic of $\mathcal{E}\!f\!f$ to translate it into recursion theory.

However, $\mathcal{E}\!f\!f$ can't do all our work for us. In particular, it doesn't seem to help us construct counterexamples to classical theorems that may fail in recursive mathematics. These counterexamples involve specifically recursion-theoretic constructions (using, e.g., Specker sequences or the priority method) rather than general constructive reasoning[10]. Even here, though, using the logic of $\mathcal{E}\!f\!f$ does give us a systematic way of putting together old counterexamples to make new ones.

Characterising ω-sets and 'real' subsets

> So, the throwing in of double negations, at first sight an easy method to create within intuitionism a counterpart to classical mathematics, is not that well understood, after all. This remark leaves us thoughtful about our own results, and renders us silent.[11]

What are ω-sets? It is possible to characterise the objects in $\mathcal{E}\!f\!f$ which come from ω-sets in purely logical terms. These descriptions are based on some remarkable results of Hyland (in [7, sections 5,6]). We will not give the proofs. Given a certain amount of topos theory, the results are almost immediate; but to prove them directly, with the tools we have to hand at present, would be extremely messy.

The following definition is not special to $\mathcal{E}\!f\!f$, but apply to any topos $\mathcal{E}$; and the result on ω-sets generalizes to any realizability topos. The nicest proof relies on the general theory of j-operators, and so is omitted here. (We could give a direct proof here, but it would be rather messy.)

Definition 4.5.20 $(X, =) \in \mathcal{E}\!f\!f$ *is* ***¬¬-separated*** *if*

$$\mathcal{E} \models \forall x, x' : X.\, \neg\neg(x = x') \rightarrow x = x'$$

[10] They necessarily do, since Bishop-style constructivism is consistent with classical mathematics.

[11] Gielen, de Swart and Veldman, 1981.

i.e. if $\delta: (X, =) \rightarrowtail (X, =) \times (X, =)$ *is* $\neg\neg$*-closed, see 4.5.23.*

(We will see again, in proposition 4.5.25, that the $\neg\neg$-operation has some mysterious role to play in the logic of $\mathcal{E}\!f\!f$.)

Proposition 4.5.21 $(X, =)$ *is* $\neg\neg$*-separated iff it's isomorphic to an* ω*-set (i.e. to an object in the image of* ω*-*$\mathcal{S}ets \hookrightarrow \mathcal{E}\!f\!f$*).*

Although we omit the proof, we should at least explain why any ω-set $(X, \in)$ is $\neg\neg$-separated as an object of $\mathcal{E}\!f\!f$. A realizer e for

$$\bigcap_{x,x' \in X} (Ex \wedge Ex' \wedge \neg\neg|x = x'|) \to |x = x'|$$

must, given any $x, x' \in X$, take codes $n \in |x \in X'|, n' \in |x' \in X'|, m \in \neg\neg|x = x'|$ and produce a code $k \in |x = x'|$. But note that $\neg\neg|x = x'|$ is nonempty exactly when $x = x'$ (in which case $\neg\neg|x = x'| = \mathbb{N}$, cf. proposition 4.1.11). So we can let $e = \Lambda n, n', m.n$ (or $\Lambda n, n', m.n'$).

Corollary 4.5.22 ω*-*$\mathcal{S}ets$ *is locally cartesian closed.*

Proof. The category of $\neg\neg$-separated objects in a topos (indeed, the category of j-separated objects for any j-operator) is, for general reasons, locally cartesian closed. ∎

Note that the definition of $\neg\neg$-separated is phrased entirely in the internal language, and so makes sense in any slice topos $\mathcal{E}\!f\!f/(X, =)$. Furthermore, pullback (reindexing) functors f^* preserve the property of being $\neg\neg$-separated, since they are logical. So we can define a full subfibrations

$$\begin{array}{c} \mathcal{S}ep \\ \downarrow \\ \mathcal{E}\!f\!f \end{array} \hookrightarrow \begin{array}{c} \mathcal{E}\!f\!f^{\to} \\ \downarrow \\ \mathcal{E}\!f\!f \end{array}$$

It turns out that $\begin{array}{c} \omega\text{-}\mathcal{S}ets^{\to} \\ \downarrow \\ \omega\text{-}\mathcal{S}ets \end{array}$ can be obtained from $\begin{array}{c} \mathcal{S}ep \\ \downarrow \\ \mathcal{E}\!f\!f \end{array}$ via change of base along the full embedding ω-$\mathcal{S}ets \hookrightarrow \mathcal{E}\!f\!f$. It is really this *fibred* version of the result that we want. The proof is easy, and combines the facts we already know with some general results on j-operators.

There is a similar characterisation of objects of the form ∇X: they are the '$\neg\neg$-sheaves'. Again, one of the reasons why this logical characterisation is useful is that it gives us a more general notion that makes sense in any slice. This is exploited in [10].

$\neg\neg$-closed predicates and subobjects There is a nice characterisation of subobjects in $\mathcal{E}\!f\!f$ that "come from real subsets". We won't discuss it in full generality here, but simply state the result for ω-sets, since it's simplest to understand in that case. We will have much more to say about this elsewhere.

Definition 4.5.23 *A predicate* $\phi: X \to \Omega$ *is* ***$\neg\neg$-closed*** *if*

$$\mathcal{E} \models \forall x : X. \neg\neg\phi(x) \to \phi(x)$$

A subobject is ***$\neg\neg$-closed*** *if its classifying map is* $\neg\neg$*-closed.*

Exercise 4.5.24 Explain why a predicate ϕ on $(X, \in)$ is $\neg\neg$-closed if and only if it's represented by a function $R: X \to \mathbf{P}\mathbb{N}$ such that for all $x \in X$, $R(x) = \varnothing$ or $\mathbb{N}$, cf. 4.1.11. ∎

Proposition 4.5.25 *Let $(X, \in)$ be an ω-set. Any $\neg\neg$-closed subobject of $(X, \in)$ is isomorphic to one of the form*

$$(X', \in) \rightarrowtail (X, \in)$$

where $X' \subseteq X$ and $|x \in X'| = |x \in X|$.

This follows from 4.5.13 and the previous exercise. The intuition is that $\neg\neg$-closed subobjects correspond to ordinary subsets, whereas in general a subobject may be specified using extra 'intensional' or 'computational' information. Taking the double negation 'kills' such information: cf. 4.1.11 and the previous exercise. The following example may also help illustrate this point.

Exercise 4.5.26 By our remarks earlier about recursion theory in $\mathcal{E}\!f\!f$, we can define an internal 'halting set' in $\mathcal{E}\!f\!f$:

$$\bar{K} = \{e : N \mid \exists y.\, T(e, e, y)\} \rightarrowtail N$$

T is decidable, so by Markov's Principle,

$$\neg\neg(\exists y.\, T(e, e, y)) \to \exists y.\, T(e, e, y)$$

i.e. $\bar{K}$ is a $\neg\neg$-closed subobject of N. As above, it comes from the ordinary halting set $K \subseteq \mathbb{N}$. Now consider the subobject

$$\bar{K} \vee \neg\bar{K} = \{e : N \mid e \in \bar{K} \vee e \notin \bar{K}\} \rightarrowtail N$$

Show that, thinking of N as an ω-set, this is just the subobject of N we get by putting $A = K$ in example 4.3.23 (one can check this using 4.5.13). Show that it is *not* $\neg\neg$-closed. (Hint: suppose it is isomorphic to one which comes from a subset $L \subseteq \mathbb{N}$ as above; show that $L = \mathbb{N}$; conclude that the halting problem is solvable.) ■

This example also shows that an epi in ω-$\mathcal{S}ets$ need not be an epi in $\mathcal{E}\!f\!f$.

4.6 The category of modest sets

We are finally in a position to describe what the 'internal' version of the PER model in $\mathcal{E}\!f\!f$ is: it is the category **Mod** of modest sets. This section defines and characterises the modest sets, describes the internal category of modest sets and states its completeness properties. The actual proofs are omitted, though we do sketch the core of an argument (due to Rosolini and Scott) that **Mod** is weakly complete. For a more elementary account of all this, see [20]. Note that all the results are valid in an arbitrary realizability topos, not just $\mathcal{E}\!f\!f$.

Note that there are still many open questions. The most important are related to the idea of *parametricity*: it turns out that PER models have surprisingly good parametricity properties, at least in the 'dinatural' sense[12]. I have not discussed such issues here.

[12] I haven't kept up with the latest results. Also, the whole question of relating the 'dinatural' approach to parametricity to the (more general, more useful, less general, more or less problematic) Reynolds or Reynolds-Ma approach seems to remain open.

PERs as modest sets

From now on we denote objects $(X,=),(I,=) \in \mathcal{E}\!f\!f$ simply by X, I.

Definition 4.6.1 *If $R : P(N \times N)$, the formula "R* ***is a closed PER****" is*

$$\text{“}R \textit{ is } \neg\neg\textit{-closed, symmetric and transitive”}$$

We let

$$\mathrm{Dom}\, R = \{n : N \mid nRn\}$$

(which is clearly a $\neg\neg$-closed subobject of N).

Definition 4.6.2 *We let*

$$\mathbf{Mod}_0 \;=\; \{R : P(N \times N) \mid \text{“}R \textit{ is a closed PER”}\} \rightarrowtail N \times N$$

In view of 4.5.25, particular relations $R \rightarrowtail N \times N$ such that

$$\mathcal{E}\!f\!f \models \text{“}R \text{ is a closed PER”}$$

correspond to ordinary[13] PERs $R \subseteq \mathbb{N} \times \mathbb{N}$. In fact, it turns out that $\mathbf{Mod}_0 = \nabla P_0$, where P_0 was the set of PERs on $\mathbb{N}$: the proof uses general properties of j-operators.

It is easy to verify that quotients of closed PERs

$$R \rightrightarrows N \twoheadrightarrow Q$$

are the same as subquotients of N

$$N \leftarrowtail A \overset{q}{\twoheadrightarrow} Q$$

where $A \rightarrowtail N$ and the kernel pair $E \rightarrowtail A \times A$ of $A \twoheadrightarrow Q$ are both $\neg\neg$-closed; i.e.

$$\begin{array}{l} \mathcal{E}\!f\!f \models \forall n : N.\, \neg\neg(n \in A) \to n \in A \\ \mathcal{E}\!f\!f \models \forall n : A.\, \neg\neg(q(n) = q(n')) \to q(n) = q(n') \end{array}$$

Definition 4.6.3 *An object $X \in \mathcal{E}\!f\!f$ is* ***modest****, or a* ***modest set*** *if*

$$\mathcal{E}\!f\!f \models \exists R : \mathbf{Mod}_0.\, X \cong N/R$$

Similarly for $(X_i) \in \mathcal{E}\!f\!f/I$.

This clearly defines a full subfibration

$$\begin{array}{ccc} \mathcal{M}od & & \mathcal{E}\!f\!f^{\to} \\ \downarrow & \hookrightarrow & \downarrow \\ \mathcal{E}\!f\!f & & \mathcal{E}\!f\!f \end{array}$$

of (families of) modest sets. This is exactly the fibration of families (X_i) such that

$$\mathcal{E}\!f\!f \models \text{“}X_i \text{ is a quotient of } N \text{ by a closed PER”}$$

The following is also easy to prove, cf. 3.8.14 and 3.8.16:

[13] This assumes that the meaning of "is a symmetric and transitive relation" is the same in $\mathcal{E}\!f\!f$ and $\mathcal{S}ets$. This can be made precise and proved: see [7, proposition 6.5] and the nonexistent notes.

Proposition 4.6.4 $(X_i) = \begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$ *is modest iff there is an epi* $\phi : K \twoheadrightarrow I$ *and a diagram*

$$K^*N \leftarrowtail \begin{pmatrix} A \\ \downarrow \\ K \end{pmatrix} \twoheadrightarrow \phi^*\begin{pmatrix} X \\ \downarrow \\ I \end{pmatrix}$$

where $\begin{pmatrix} A \\ \downarrow \\ K \end{pmatrix} \rightarrowtail K^*N$ *and the kernel pair* $E \rightarrowtail \begin{pmatrix} A \\ \downarrow \\ K \end{pmatrix} \times_K \begin{pmatrix} A \\ \downarrow \\ K \end{pmatrix}$ *are* $\neg\neg$*-closed.*

That is, X is modest iff its "locally" a subquotient of N by a closed PER.

Some basic properties The following are proved in [10], using an idea of Freyd's:

- $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$ is a strongly complete fibration, inheriting its limits from $\begin{array}{c}\mathcal{E}ff^{\to}\\ \downarrow\\ \mathcal{E}ff\end{array}$;
- $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$ is a strongly cocomplete fibration (Σ's computed by composition, as in $\begin{array}{c}\mathcal{E}ff^{\to}\\ \downarrow\\ \mathcal{E}ff\end{array}$);
- modest families over ω-sets I are genuine quotients of N by closed PERs $R \rightarrowtail I^*(N \times N)$;
- hence $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ is obtained from $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$ via change of base along $\omega\text{-}\mathcal{S}ets \hookrightarrow \mathcal{E}ff$.

Note that we have the **generic quotient of N by a closed PER** or **generic modest set**:

$$\begin{pmatrix} G \\ \downarrow \\ \mathbb{M}\mathbf{od}_0 \end{pmatrix}$$

This is not quite a generic object for $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$, but it *is* a generic object for $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ (and of course, isomorphic to the one we defined for $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \mathcal{S}ets\end{array}$). Continuing, then, we have

- $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ has a generic object;
- $\begin{array}{c}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{array}$ is a strongly complete fibration; hence
- we get a model for the theory of constructions, with strong sums.

Moreover, we can talk about this model in the internal language of $\mathcal{E}ff$: it is 'set-theoretic'.

Exercise 4.6.5 Define the notion of 'weak' generic object, analogously to the notion of weak completeness for a fibration, and show that the generic modest set is a weak generic object for the fibration $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$. ∎

Unlike $\begin{array}{c}\mathcal{M}od\\ \downarrow\\ \mathcal{E}ff\end{array}$, the fibration of genuine subquotients of N (cf. 3.8.2) is only weakly complete over $\mathcal{E}ff$; it is not a trivial fact that it is strongly complete when restricted to a fibration over ω-$\mathcal{S}ets$.

Fullness Actually, we have glossed over something here. We know that the *objects* of the fibration are the same; but what about the *arrows*? There seems to be no reason why the arrows between two PERs in Per (or between two families of PERs in $\begin{smallmatrix}\mathbb{P}\\ \downarrow \\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$) should be the same as the arrows between the corresponding modest sets in $\mathcal{E}\!f\!f$. And if this fails, the interpretations of function types will differ. So we'd better sort this out.

Definition 4.6.6 *Let R, S be closed PERs. Define a new closed PER S^R by putting*

$$\begin{array}{rcl} \mathrm{Dom}\, S^R & = & \{e : N \mid \forall n, n' : N.\, nRn' \to \{e\}(n)S\{e\}(n')\} \\ e(S^R)e' & \textit{iff} & \forall n : N.\, n \in \mathrm{Dom}\, R \to \{e\}(n)S\{e'\}(n) \end{array}$$

(The fact that it's a closed PER is a simple exercise in logic.)

This is the same definition we had for ordinary PERs (see 4.2.4), only phrased in the internal language. Clearly there's an arrow in $\mathcal{E}\!f\!f$ that takes an equivalence class $p = [e]$ of codes to the function f_p they all track:

$$N/(S^R) \longrightarrow (N/S)^{(N/R)} \qquad p \mapsto f_p$$

We want to show this is an isomorphism. This follows from a version of Extended Church's Thesis. Since we want the argument to apply to any realizability topos, we must formulate it without using T's or U's. So we note that (Kleene) application defines a partial map

$$N \times N \rightharpoonup N \qquad e, n \mapsto e \cdot n$$

(i.e. $e \cdot n = \{e\}(n)$.) A corresponding partial map exists in any realizability topos.

It is simpler to work, not with PERs, but with subquotients: so suppose we have

$$\begin{array}{c} N \leftarrowtail A \overset{q_P}{\twoheadrightarrow} P \\ N \leftarrowtail B \overset{q_Q}{\twoheadrightarrow} Q \end{array}$$

where $A, B \rightarrowtail N$ are $\neg\neg$-closed.

Lemma 4.6.7 *(Cf. [7, proposition 10.4])*

$$\mathcal{E}\!f\!f \models \forall f : Q^P.\, \exists e : N.\, \forall n : A.\, f(q_P(n)) = q_Q(e \cdot n)$$

Proof. Similar to Church's Thesis, but harder. ∎

This is valid in any realizability topos. In fact,

Lemma 4.6.8 *The following universally quantified formula holds:*

$$\forall R, S : \mathbf{Mod}_0.\, \forall f : (N/S)^{(N/R)}.\, \exists e : N.\, \forall n : \mathrm{Dom}\, R.\, f([n]_R) = [e \cdot n]_S$$

Proof. Because the previous proof is 'uniform' in P, Q. ∎

It follows that lemma 4.6.7 is true *in any slice of $\mathcal{E}\!f\!f$*, i.e. for subquotients of N in any slice. Note that to prove this directly, in each slice, would require a lot more tedious calculations.

Proposition 4.6.9 *If R, S are closed PERs, the arrow $p \mapsto f_p$ defines an isomorphism*

$$N/(S^R) \cong (N/S)^{(N/R)}$$

Proof. Let $P = N/R, Q = N/S$ and let $f : Q^P$. Suppose we have $e : N$ as in the lemma; then $e \in \mathrm{Dom}\, S^R$ and if p is the S^R-equivalence class of e, $f = f_p$. Moreover, it does not matter which e we choose. Therefore, taking the S^R-equivalence class p of e, we have

$$\mathcal{E}\!f\!f \models \forall f : Q^P.\, \exists! p : N/(S^R).\, f = f_p$$

Thus we have an inverse $Q^P \to N/(S^R)$, namely

$$\lambda f.\, \mathsf{the}\, p\, \mathsf{such\ that}\ f = f_p$$

■

It is now nearly immediate that any arrow $(N/R) \to (N/S)$ in $\mathcal{E}\!f\!f$ comes from an arrow in Per, and similarly for any fibre of $\begin{smallmatrix}\mathbb{P}\\ \downarrow\\ \omega\text{-}\mathcal{S}ets\end{smallmatrix}$. This plugs the gap we mentioned.

Corollary 4.6.10 *If X, Y are modest, then so is Y^X.*

Proof. We have just shown that if $X \cong N/R$ and $Y \cong NS$ then $Y^X \cong N/(S^R)$. ■

Note that this argument is valid in arbitrary slices of $\mathcal{E}\!f\!f$, i.e. for arbitrary *families* of modest sets.

Products as intersections There is something else we have glossed over. Since indexed products in $\begin{smallmatrix}\mathcal{M}od\\ \downarrow\\ \mathcal{E}\!f\!f\end{smallmatrix}$ coincide with those in $\begin{smallmatrix}\mathcal{E}\!f\!f^{\to}\\ \downarrow\\ \mathcal{E}\!f\!f\end{smallmatrix}$, if we use it to model polymorphic type theories then Π-types are interpreted as 'set-theoretic' products: e.g.

$$[\![\Pi X.X \to X]\!] = \prod_{R:\mathbf{Mod}_0} (N/R)^{(N/R)}$$

There seems to be no reason why this should coincide with the quotient

$$N/\Big(\bigcap_{R:\mathbb{M}\mathbf{od}_0} R^R\Big)$$

i.e. with our previous interpretation of polymorphic types in Per.

In fact it does. There is a direct proof in [7], but it actually follows from the uniformity principle. The following argument is due to Rosolini and Scott. It clearly generalises to more complicated types. First, note that one can define the intersection of a family $(R_i)_{i:I}$ of closed PERs as

$$\bigcap_{i:I} R_i = \{\langle n, n'\rangle : N \times N \,|\, \forall i : I.\, nR_i n'\}$$

and show that this is again a closed PER.

Proposition 4.6.11 *In $\mathcal{E}\!f\!f$, there is an isomorphism*

$$\prod_{R:\mathbf{Mod}_0} (N/R)^{(N/R)} \cong N/\Big(\bigcap_{R:\mathbb{M}\mathbf{od}_0} R^R\Big)$$

Proof. Given $p : N/\left(\bigcap_{R:\mathbb{M}\mathbf{od}_0} R^R\right)$, let $e : N$ be a representative for p and let p_R be the R-equivalence class of e; p_R does not depend on the choice of e. Then we clearly have $f_{p_R} : (N/R)^{(N/R)}$ for all R, so we get

$$\langle R\rangle f_{p_R} : \prod_{R:\mathbf{Mod}_0} (N/R)^{(N/R)}$$

This gives us a map $p \mapsto \langle R\rangle f_{p_R}$ in one direction.

On the other hand, suppose $f : \prod_{R:\mathbb{M}\mathbf{od}_0}(N/R)^{(N/R)}$; then for each $R \in \mathbf{Mod}_0$, $\pi_R f : (N/R)^{(N/R)}$ and so we have $e_R \in N$ such that $\pi_R f = [e_R]$. Spelling this out in the internal language, we have

$$\forall R : \mathbf{Mod}_0.\, \exists e : N.\, \pi_R f = [e]$$

Now, since $\mathbf{Mod}_0$ is (isomorphic to) an object of the form ∇X, by 4.5.17 the uniformity principle remains valid when we replace PN with $\mathbf{Mod}_0$. So we deduce that

$$\exists e : N.\, \forall R : \mathbf{Mod}_0.\, \pi_R f = [e]$$

Since $[e] \in R^R$ for all R, this shows that

$$\forall f : \prod_{R:\mathbf{Mod}_0} (N/R)^{(N/R)}.\, \exists e : \mathrm{Dom}\left(\bigcap_{R:\mathbf{Mod}_0} R^R\right).\, \forall R : \mathbf{Mod}_0.\, \pi_R f = [e]$$

Now fix f again and suppose we have e, e' such that

$$\forall R : \mathbf{Mod}_0.\, \pi_R f = [e] = [e']$$

Then clearly $e(R^R)e'$ for all R, and so

$$e\Big(\bigcap_{R:\mathbb{M}\mathbf{od}_0} R^R\Big)e'$$

So we conclude that

$$\forall f : \prod_{R:\mathbb{M}\mathbf{od}_0} (N/R)^{(N/R)}.\, \exists! p : N/\Big(\bigcap_{R:\mathbf{Mod}_0} R^R\Big).\, \forall R : \mathbf{Mod}_0.\, \pi_R f = p$$

(i.e. putting $p = [e]$.) Hence we have a map in the other direction, namely

$$\lambda f.\, \mathsf{the}\ p\ \mathsf{such\ that}\ \forall R : \mathbf{Mod}_0.\, \pi_R f = p$$

And this is clearly inverse. ∎

This really amounts to a proof that the product is a modest set. In fact, one can use this as a basis for a completely internal proof that the internal category **Mod** (shortly to be defined) is weakly complete: Rosolini has carried this out. However, one's still left with the problem of establishing the 'strong' completeness properties over *ω-Sets* that are needed when modelling type theories. This requires some extra work.

Interestingly, the uniformity principle seems to have motivated [10].

The internal category of modest sets

For those who prefer modelling their polymorphic type theories using internal categories...

Definition 4.6.12 Mod *is the internal full subcategory (cf. definition 3.8.27) on the generic modest set (see 4.3.20)*

$$\begin{pmatrix} G \\ \downarrow \\ \mathbb{M}\boldsymbol{od}_0 \end{pmatrix}$$

i.e. on the family of quotients of N by closed PERs.

By the earlier description of exponentials of modest sets, together with results from the theory of j-operators, $\mathbf{Mod}_1$ is an ω-set: [10, lemma 7.8]. So **Mod** is an internal category in ω-$\mathcal{S}ets$, not just $\mathcal{E}\!f\!f$.

The following completeness properties are proved in [10]. They are a consequence of the results on fibrations we listed earlier, cf. 3.8.40 and [10].

- **Mod** is weakly complete;
- **Mod** is strongly complete over any ω-set I; and
- $\begin{matrix} \mathcal{M}od \\ \downarrow \\ \mathcal{E}ff \end{matrix}$ is 'weakly' equivalent to the externalisation of **Mod**.

What more could you ask?[14]

Subcategories of modest sets

Often we want to study modest sets with particular properties. That is, we want to study a full subfibration of $\begin{matrix} \mathcal{M}od \\ \downarrow \\ \mathcal{E}ff \end{matrix}$, or a full subcategory of **Mod**.

Suppose we have an internal collection $C_0 \rightarrowtail \mathbf{Mod}_0$ of closed PERs, classified by (say) $\phi\colon \mathbf{Mod}_0 \to \Omega$, where ϕ is the property we're interested in. Form the pullback

$$\begin{tikzcd} H \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & G \arrow[d] \\ C_0 \arrow[r] & \mathbf{Mod}_0 \end{tikzcd}$$

Then $\begin{pmatrix} H \\ \downarrow \\ C_0 \end{pmatrix} \in \mathcal{E}/C_0$ is the **generic modest set with property** ϕ. We can use it to define a full subfibration

$$\begin{matrix} \mathcal{C} \\ \downarrow \\ \mathcal{E}ff \end{matrix} \hookrightarrow \begin{matrix} \mathcal{M}od \\ \downarrow \\ \mathcal{E}ff \end{matrix}$$

[14] (rhetorical)

as in proposition 3.8.9, or we can take the internal full subcategory on the family $\begin{pmatrix} H \\ \downarrow \\ C_0 \end{pmatrix}$ to get an internal subcategory

$$\mathbf{C} \hookrightarrow \mathbf{Mod}$$

This is the **full subcategory of Mod on the objects** C_0. $\mathbf{C}$ and $\begin{matrix} \mathcal{C} \\ \downarrow \\ \mathcal{E}\!f\!f \end{matrix}$ have the same relationship as $\mathbf{Mod}$ and $\begin{matrix} \mathcal{M}od \\ \downarrow \\ \mathcal{E}\!f\!f \end{matrix}$.

Back to decidable PERs Here's an example. Recall from exercise 4.3.15 that we succeeded in constructing a fibration of decidable PERs over ω-*Sets* that had a generic object. We will show how this arises in the way we have just described.

Definition 4.6.13 *An object $X \in \mathcal{E}\!f\!f$ (or in any topos) is **decidable** if*

$$\mathcal{E}\!f\!f \models \forall x, x' \in X. (x = x') \vee (x \neq x')$$

Using this idea, we can define the internal object of decidable PERs:

Definition 4.6.14 *(Cf. 4.5.10) Let*

$$D_0 = \{R \in P_0 | \, \forall x, x' \in \mathbb{N}/R. (x = x') \vee (x \neq x')\}$$

and let $\mathbf{D}$ *be the full subcategory of* $\mathbf{Mod}$ *on* D_0*.*

Note that $D_0 \rightarrowtail \mathbf{Mod}_0$ is *not* $\neg\neg$-closed.

Proposition 4.6.15 *This is isomorphic to the ω-set D_0 defined in 4.3.15.*

Proof. Recall that the underlying set of $\mathbf{Mod}_0$ is P_0. One checks that the predicate

$$\text{“}\forall x, x' \in \mathbb{N}/R. (x = x') \vee (x \neq x')\text{”}$$

is represented by the function $P_0 \to \mathbf{P}\mathbb{N}$ sending

$$R \mapsto \{e \in \mathbb{N} | \, e \text{ is a decision code for } R\}$$

The result follows from 4.5.13. ∎

The internal category $\mathbf{D}$ of decidable modest sets is not complete, of course: for example, N is decidable but $\prod_{n:N} = N^N$ isn't, as there is no effective procedure for deciding whether two total recursive functions are equal.

The advantage of having an *internal* definition of decidable modest set, rather than our previous recursion-theoretic definition of decidable PER, is that it is much easier to verify that something is a decidable modest set. For example, it is almost immediate that algebraic data types constructed from decidable objects are again decidable, whereas checking the equivalent fact for decidable PERs is a little tedious.

Completeness of subcategories Here we make some simple remarks about full subcategories $\mathbf{C} \hookrightarrow \mathbf{Mod}$ that are *complete*, or more precisely, closed under limits in **Mod** (i.e. limits in $\mathcal{E}\!f\!f$, in whatever sense they exist in **Mod**). We give no examples of such categories; the most interesting are categories of so-called *synthetic domains*, studied elsewhere (e.g. [16]).

Clearly **C** must be cartesian closed. For

$$Y^X = \prod_{x:X} Y$$

The second is a remark little trickier. Even if **C** is complete, the analogue of proposition 4.6.11 may fail: we may not be able to calculate C_0-indexed products in **C** as C_0-indexed intersections of PERs. For example,

$$\prod_{R:C_0} (N/R)^{(N/R)} \;\cong\; N/\Big(\bigcap_{R:C_0} R^R\Big)$$

need not hold. The reason is that the proof of 4.6.11 used the uniformity principle for $\mathbf{Mod}_0$; and there is no reason why it should hold for C_0, unless C_0 is of the form ∇X. (But Freyd has conjectured that **C** will always be equivalent to an internal category with an object-of-objects of this form.)

However, one should not be deceived into thinking that, simply because products cannot obviously be calculated as intersections, that they cannot exist. And one should be wary of arguments which suggest that C_0-indexed products can't exist on the grounds of 'uniformity': the point is that we are calculating a product indexed, not by a set, but by an ω-set, and the extra intensional information contained in each $|c \in C_0|$ may be just what we need to be able to verify that the product, which certainly lies in **Mod**, actually lies in **C**.

Reflectivity We mention the following result (whose converse is obvious):

Proposition 4.6.16 *If* **C** *is closed under limits in* **Mod***, there is an internal reflection*

$$F\colon \mathbf{Mod} \to \mathbf{C}$$

Proof. This is just Freyd's adjoint functor theorem for internal categories. There is one subtlety: since all the limits required by that theorem are indexed by internal categories in ω-$\mathcal{S}ets$, they exist in **Mod** in the strong, and not just the weak, sense; that's because **Mod** is *strongly*, and not just weakly, complete over an ω-set. This is why we can show that there is actually an internal reflection functor, rather than just being able to prove

$$\mathcal{E}\!f\!f \models \exists F\colon \mathbf{Mod} \to \mathbf{C}.\ \text{“}F \text{ is a reflection”}$$

The task of fleshing out this argument is left as an exercise (recall that the reflection functor takes $X \in \mathbf{Mod}$ to an initial object for the comma category $(\ulcorner X \urcorner \downarrow I)$, where $I\colon \mathbf{C} \hookrightarrow \mathbf{Mod}$.) ∎

Note that

- If we want to construct an actual reflection functor, we cannot argue *entirely* in the internal language; and
- Although we have arrows $\mathbf{C}_0 \rightarrowtail \mathbf{Mod}_0 \to \mathbf{C}_0$, we do *not* know that this is a retraction pair; only that it takes any $X \in \mathbf{C}_0$ to something *isomorphic*.

Freyd has conjectured that one can always obtain a retraction; the conjecture mentioned earlier would be a consequence.

Reflectivity has many useful consequences in synthetic domain theory: for example, it implies the existence of convex powerdomains (i.e. free semilattices; cf. [16])—whereas to describe them explicitly as PERs would be extremely hard. In fact, the approach does not even makes sense unless we are prepared to take the plunge and work with categories relative to a base topos, rather than just calculating with PERs.

Bibliography

[1] M. Beeson: *Foundations of Constructive Mathematics*, Springer, 1985.

[2] J. L. Bell: *Toposes and Local Set Theories*, Oxford Logic Guide 16, 1988.

[3] J. Bénabou: Fibred categories and the foundations of naive category theory, J. Symb. Logic **50** (1985), 10–37.

[4] A. Carboni, P. J. Freyd, A. Scedrov: A categorical approach to realizability and polymorphic types, in *Mathematical Foundations of Programming Language Semantics* (eds. M. Main, A. Melton, M. Mislove, D. Schmidt), Springer LNCS 298, 1988.

[5] M. P. Fourman: The logic of topoi, in *Handbook of Mathematical Logic* (ed. J. Barwise), Studies in Logic and the Foundations of Mathematics 90, North-Holland, 1977.

[6] R. Goldblatt: *Topoi: The Categorial Analysis of Logic*, Studies in Logic and the Foundations of Mathematics 98, North-Holland, 1979.

[7] J. M. E. Hyland: The effective topos, in: *The L. E. J. Brouwer Centenary Symposium* (ed. A. S. Troelstra, D. van Dalen), North-Holland, 1982.

[8] J. M. E. Hyland: A small complete category, *Annals of Pure and Applied Logic* 40 (1988) 135-165.

[9] J. M. E. Hyland, P. T. Johnstone, A. M. Pitts: Tripos theory, *Math. Proc. Camb. Phil. Soc.* (1980), **88**, 205.

[10] J. M. E. Hyland, E. P. Robinson, G. Rosolini: The discrete objects in the effective topos, *Proc. London Math. Soc.* **60** (1990), 1–36.

[11] B. P. F. Jacobs: *Categorical Type Theory*, thesis, University of Nijmegen, 1991.

[12] P. T. Johnstone: *Topos Theory*, LMS Mathematical Monographs 10, Academic Press, 1977.

[13] J. Lambek, P. J. Scott: *Introduction to Higher-Order Categorical Logic*, Cambridge Studies in Advanced Mathematics 7, 1986.

[14] D. Lehmann, M. Smyth: Algebraic specification of data types, a synthetic approach, *Math. Systems Theory 14* (1981).

[15] G. Longo, E. Moggi: Constructive natural deduction and its "ω-set" interpretation, LIENS–90–21, October 1990; to appear in *Math. Structures in Comp. Sci.*

[16] W. K.-S. Phoa: Effective domains and intrinsic structure, in: *Proc. of the Fifth IEEE Symposium on Logic in Computer Science*, IEEE Computer Society Press, 1990.

[17] A. M. Pitts: Polymorphism is set-theoretic, constructively, in: *Category Theory and Computer Science* (eds. D. H. Pitt *et al.*), Lecture Notes in Computer Science 283, Springer, 1987.

[18] W. Powell: manuscript, 1976.

[19] J. C. Reynolds: Polymorphism is not set-theoretic, in: *Semantics of Data Types* (eds. G. Kahn *et al*), Lecture Notes in Computer Science 173, Springer, 1984.

[20] E. Robinson: How complete is PER? in: *Proc. of the Fourth IEEE Symposium on Logic in Computer Science*, IEEE Computer Society Press, 1989.

[21] G. Rosolini: About modest sets, to appear.

[22] R. A. G. Seely: Locally cartesian closed categories and type theory, *Math. Proc. Camb. Phil. Soc.* **95** (1984) 33–48.

[23] R. A. G. Seely: Categorical semantics for higher-order polymorphic λ-calculus, *J. Symb. Logic* **52** (1987) 969–989.

Appendices

I: A version of constructive set theory

There are four kinds of judgements. They all have the form

$$[\Gamma]\ \cdots$$

where Γ is an *environment*; i.e. a list of variable declarations

$$x_1 : X_1, \ldots, x_n : X_n$$

where the X_i are expressions denoting sets. It's often convenient to write "$x, x' : X$" instead of "$x : X, x' : X$" in an environment, though we won't do that below.

The different kinds of judgements are:

1. Ones that allow us to write down expressions which denote sets. These judgements are of the form
$$[\Gamma]\ X \text{ set}$$
where X is an expression denoting a set.

2. Ones that allow us to write down expressions which denote elements of a set; that is, *terms*. These judgements are of the form
$$[\Gamma]\ t : X$$
where X is an expression denoting a set and t is a term.

3. Ones that tell us when two terms describe the same element of a set. These judgements are of the form
$$[\Gamma]\ t = t' : X$$
where X is an expression denoting a set and t, t' are terms. This is a kind of 'external' equality; if the rules are read left-to-right (as in $t \to t'$), they can be viewed as giving a rewrite system that allows us to 'evaluate' terms.

4. Ones that tell us what logical deductions, involving set-theoretic formulas, are valid. These judgements are of the form
$$[\Gamma]\ \vec{\phi} \vdash \psi$$
where ψ is a predicate (i.e. a term of type Ω) and
$$\vec{\phi} = \phi_1, \ldots, \phi_k$$
is a list of hypotheses: each ϕ_i is a predicate. (By convention, Greek letters ϕ, ψ denote terms of type Ω, i.e. truth-values.)

There are a number of rules that tell us which judgements are derivable. There is some redundancy built into the rules; and notational consistency is traded off against readability. Smaller, more elegant systems exist, but they are more work to use in practice.

If $[\Gamma, x : X]\ Y$ set, we often write "$Y(x)$" instead of Y to emphasise that Y may contain the free variable x. If, in addition, $[\Gamma]\ t : X$, we write "$Y(t)$" instead of $Y[x := t]$. Similarly for predicates: we write $\phi(x), \phi(t)$ instead of $\phi, \phi[x := t]$.

Our system does not have explicit proof-terms $p : \phi$. Toposes model provability, not proofs; all such proof-terms would be identified in a topos model. However, a system with proof-terms might be useful in studying realizability toposes.

General structural rules The following rules (exchange, weakening and substitution for a variable) apply to all four kinds of judgement. Here the ellipsis "$\cdots$" denotes the conclusion of a judgement.

$$\frac{[\Gamma, x : X, y : Y, \Gamma']\ \cdots}{[\Gamma, y : Y, x : X, \Gamma']\ \cdots}\ (\text{provided } x \notin FV(Y)) \qquad \frac{[\Gamma]\ X \text{ type} \quad [\Gamma]\ \cdots}{[\Gamma, x : X]\ \cdots}$$

$$\frac{[\Gamma, x : X, \Gamma']\ \cdots \quad [\Gamma]\ t : X}{[\Gamma, \Gamma'[x := t]]\ (\cdots)[x := t]}$$

The last rule is probably derivable, but it's handy to write it down explicitly.

1: Rules for defining sets

Basic sets

$$\frac{}{[\Gamma]\ 1 \text{ set}} \qquad \frac{}{[\Gamma]\ \Omega \text{ set}}$$

Cartesian products, disjoint unions and function spaces

$$\frac{[\Gamma]\ X \text{ set} \quad [\Gamma]\ Y \text{ set}}{[\Gamma]\ X \times Y \text{ set}} \qquad \frac{[\Gamma]\ X \text{ set} \quad [\Gamma]\ Y \text{ set}}{[\Gamma]\ X + Y \text{ set}}$$

"2" is syntactic sugar for $1 + 1$, "3" for $1 + 1 + 1$, and so on.

$$\frac{[\Gamma]\ X \text{ set} \quad [\Gamma]\ Y \text{ set}}{[\Gamma]\ Y^X \text{ set}}$$

"PX" is syntactic sugar for Ω^X.

Subsets formed by separation

$$\frac{[\Gamma, x : X]\ \phi(x) : \Omega}{[\Gamma]\ \{x : X \mid \phi(x)\} \text{ set}}$$

This is the basic way that dependent types can arise.

Indexed products and coproducts

$$\frac{[\Gamma, x : X]\ Y(x) \text{ set}}{[\Gamma]\ \prod_{x:X} Y(x) \text{ set}} \qquad \frac{[\Gamma, x : X]\ Y(x) \text{ set}}{[\Gamma]\ \coprod_{x:X} Y(x) \text{ set}}$$

These correspond to dependent products and 'strong sums' in Martin-Löf type theory.

2: Rules for writing down terms

Introducing a variable

$$\frac{[\Gamma]\ X\ \text{set}}{[\Gamma, x : X]\ x : X}$$

Basic sets

$$\frac{}{[\Gamma]\ * : 1}$$

$$\frac{}{[\Gamma]\ \mathsf{true} : \Omega} \qquad \frac{}{[\Gamma]\ \mathsf{false} : \Omega} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ t' : X}{[\Gamma]\ \mathsf{Eq}(t, t') : \Omega}$$

We often write "$t = t'$" for $\mathsf{Eq}(t, t')$ if there is no risk of confusion. However, we will not do this when stating the rules that follow.

$$\frac{[\Gamma]\ \phi : \Omega \quad [\Gamma]\ \psi : \Omega}{[\Gamma]\ \phi \wedge \psi : \Omega} \qquad \frac{[\Gamma]\ \phi : \Omega \quad [\Gamma]\ \psi : \Omega}{[\Gamma]\ \phi \vee \psi : \Omega} \qquad \frac{[\Gamma]\ \phi : \Omega \quad [\Gamma]\ \psi : \Omega}{[\Gamma]\ \phi \rightarrow \psi : \Omega}$$

If $[\Gamma]\ \phi : \Omega$, "$\neg\phi$" is syntactic sugar for $\phi \rightarrow \mathsf{false}$.

$$\frac{[\Gamma, x : X]\ \phi(x) : \Omega \quad [\Gamma, x' : \{x : X \mid \phi(x)\}]\ \psi(x') : \Omega}{[\Gamma, x : X]\ \phi(x)\ \mathsf{andalso}\ \psi(x) : \Omega}$$

The above rule is rarely used explicitly.

$$\frac{[\Gamma, x : X]\ \phi(x) : \Omega}{[\Gamma]\ \forall x : X.\, \phi(x) : \Omega} \qquad \frac{[\Gamma, x : X]\ \phi(x) : \Omega}{[\Gamma]\ \exists x : X.\, \phi(x) : \Omega}$$

We often write "$\forall x, x' : X.\, \phi(x, x')$" instead of $\forall x : X.\, \forall x' : X.\, \phi(x, x')$; similarly for $\exists$.

Cartesian products, disjoint unions and function spaces

$$\frac{[\Gamma]\ x : X \quad [\Gamma]\ y : Y}{[\Gamma]\ \langle t, t' \rangle : X \times Y} \qquad \frac{[\Gamma]\ t : X \times Y}{[\Gamma]\ \pi_0 t : X} \qquad \frac{[\Gamma]\ t : X \times Y}{[\Gamma]\ \pi_1 t : Y}$$

$$\frac{[\Gamma]\ t : X}{[\Gamma]\ \mathsf{in}_0 t : X + Y} \qquad \frac{[\Gamma]\ t : Y}{[\Gamma]\ \mathsf{in}_1 t : X + Y} \qquad \frac{[\Gamma]\ t : X + Y \quad [\Gamma, x : X]\ u : Z \quad [\Gamma, y : Y]\ u' : Z}{[\Gamma]\ \mathsf{case}(t, x^X \triangleright u, y^Y \triangleright u') : Z}$$

Note that the variables x and y are *bound* in the term $\mathsf{case}(t, x^X \triangleright u, y^Y \triangleright u')$. We often leave off the type annotation, writing "x" and "y" instead of x^X and y^Y when there is no risk of ambiguity. We will do this in the rules that follow.

$$\frac{[\Gamma]\ Y\ \text{set} \quad [\Gamma, x : X]\ t : Y}{[\Gamma]\ \lambda x^X.\, t : Y^X} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ f : Y^X}{[\Gamma]\ f(t) : Y}$$

In the following rules, for the sake of readability, the type annotation on the variable x will be omitted.

Also, if $[\Gamma]\ p : PX$, "$x \in p$" is syntactic sugar for $p(x)$. Furthermore, "$\{x \in X \mid \phi(x)\}$" will stand for for $\lambda x.\, \phi(x)$, and "$\varnothing$" will stand for $\{x \in X \mid \mathsf{false}\}$. However, we will not use any of this notation in the rules that follow.

Subsets formed by separation

$$\frac{[\Gamma]\ t : \{x : X \,|\, \phi(x)\}}{[\Gamma]\ \mathsf{m}_\phi t : X} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma, x : X]\ \phi(x) : \Omega \quad [\Gamma]\ \vdash \phi(t)}{[\Gamma]\ \mathsf{p}_\phi t : \{x : X \,|\, \phi(x)\}}$$

When there is no risk of confusion, we'll simply write "t" instead of $\mathsf{m}_\phi t$ or $\mathsf{p}_\phi t$. However, we won't do this when stating the rules below.

Indexed products and coproducts

$$\frac{[\Gamma, x : X]\ t : Y(x)}{[\Gamma]\ \langle x^X \rangle t : \prod_{x:X} Y(x)} \qquad \frac{[\Gamma]\ t : \prod_{x:X} Y(x) \quad [\Gamma]\ s : X}{[\Gamma]\ \pi_s t : Y(s)}$$

Note that the variable x is *bound* in the term $\langle x^X \rangle t$. In the following rules, we will omit the type annotation.

$$\frac{[\Gamma]\ s : X \quad [\Gamma]\ t : Y(s)}{[\Gamma]\ \mathsf{in}_s t : \coprod_{x:X} Y(x)} \qquad \frac{[\Gamma]\ t : \coprod_{x:X} Y(x)}{[\Gamma]\ \mathsf{which}(t) : X} \qquad \frac{[\Gamma]\ t : \coprod_{x:X} Y(x)}{[\Gamma]\ \mathsf{what}(t) : Y(\mathsf{which}(t))}$$

Definite descriptions

$$\frac{[\Gamma]\ \vdash \exists x : X.\, \phi(x) \quad [\Gamma]\ \vdash \forall x, x' : X.\, \phi(x) \wedge \phi(x') \rightarrow \mathsf{Eq}(x, x')}{[\Gamma]\ \mathsf{the}\ x^X \mathsf{such\ that}\ \phi(x) : X}$$

Here, $\mathsf{the}\ x^X \mathsf{such\ that}\ \phi(x)$ intuitively means "the unique x such that $\phi(x)$".

3: Rules for equality of terms

General rules

$$\frac{[\Gamma]\ X \text{ set}}{[\Gamma, x : X]\ x = x : X}$$

The following rule is called 'substituting equals for equals'.

$$\frac{[\Gamma, \Gamma']\ t : Y \quad [\Gamma, \Gamma']\ t' : Y \quad [\Gamma, \Gamma'[\vec{x} := \vec{s}]]\ t[\vec{x} := \vec{s}] = t'[\vec{x} := \vec{s}] : Y \quad [\Gamma]\ s_i = u_i : X}{[\Gamma, \Gamma'[\vec{x} := \vec{u}]]\ t[\vec{x} := \vec{u}] = t'[\vec{x} := \vec{u}] : Y}$$

In this rule, $\vec{x}$ is a list of variables and $\vec{t}, \vec{u}$ are lists of terms. The precise definition of substitution is left as an exercise.

The symmetry and transitivity of equality are derived rules.

Basic sets

$$\frac{[\Gamma]\ t : 1}{[\Gamma]\ t = * : 1}$$

$$\frac{[\Gamma]\ \phi : \Omega \quad [\Gamma]\ \psi : \Omega \quad [\Gamma]\ \phi \vdash \psi \quad [\Gamma]\ \psi \vdash \phi}{[\Gamma]\ \phi = \psi : \Omega}$$

Cartesian products, disjoint unions and function spaces

$$\frac{[\Gamma]\ t : X \quad [\Gamma]\ t' : Y}{[\Gamma]\ \pi_0\langle t, t'\rangle = t : X} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ t' : Y}{[\Gamma]\ \pi_1\langle t, t'\rangle = t' : Y}$$

$$\frac{[\Gamma]\ t : X \times Y}{[\Gamma]\ \langle \pi_0 t, \pi_1 t\rangle = t : X \times Y}$$

$$\frac{[\Gamma]\ t : X \quad [\Gamma, x : X]\ u : Z \quad [\Gamma, y : Y]\ u' : Z}{[\Gamma]\ \mathsf{case}(\mathsf{in}_0 t, x \triangleright u, y \triangleright u') = u[x := t] : Z} \qquad \frac{[\Gamma]\ t : Y \quad [\Gamma, x : X]\ u : Z \quad [\Gamma, y : Y]\ u' : Z}{[\Gamma]\ \mathsf{case}(\mathsf{in}_1 t, x \triangleright u, y \triangleright u') = u'[y := t] : Z}$$

$$\frac{[\Gamma, z : X + Y]\ t : Z'}{[\Gamma, z : X + Y]\ t = \mathsf{case}(z, x \triangleright t[z := \mathsf{in}_0 x], y \triangleright t[z := \mathsf{in}_1 y]) : Z'}$$

$$\frac{[\Gamma]\ \lambda x.\, t : Y^X \quad [\Gamma]\ u : X}{[\Gamma]\ (\lambda x.\, t)(u) = t[x := u] : Y} \qquad \frac{[\Gamma]\ f : Y^X}{[\Gamma]\ \lambda x.\, f(x) = f : Y^X}$$

Subsets formed by separation

$$\frac{[\Gamma, x : X]\ \phi(x) : \Omega}{[\Gamma, x' : \{x : X \mid \phi(x)\}]\ \mathsf{p}_\phi \mathsf{m}_\phi x' = x' : \{x : X \mid \phi(x)\}} \qquad \frac{[\Gamma, x : X]\ \phi(x) : \Omega \quad [\Gamma]\ t : X \quad [\Gamma]\ \vdash \phi(t)}{[\Gamma]\ \mathsf{m}_\phi \mathsf{p}_\phi t = t : X}$$

Indexed products and coproducts

$$\frac{[\Gamma, x : X]\ t : Y(x) \quad [\Gamma]\ s : X}{[\Gamma]\ \pi_s\langle x\rangle t = t[x := s] : Y(s)} \qquad \frac{[\Gamma]\ t : \prod_{x:X} Y(x)}{[\Gamma]\ \langle x\rangle \pi_x t = t : \prod_{x:X} Y(x)}$$

$$\frac{[\Gamma]\ s : X \quad [\Gamma]\ t : Y(s)}{[\Gamma]\ \mathsf{which}(\mathsf{in}_s t) = s : X} \qquad \frac{[\Gamma]\ s : X \quad [\Gamma]\ t : Y(s)}{[\Gamma]\ \mathsf{what}(\mathsf{in}_s t) = t : Y(s)} \qquad \frac{[\Gamma]\ t : \coprod_{x:X} Y(x)}{[\Gamma]\ \mathsf{in}_{\mathsf{which}(t)}\mathsf{what}(t) = t : \coprod_{x:X} Y(x)}$$

4: Rules for making logical inferences

Basic rules

$$\frac{[\Gamma, x : X]\ \phi(x) : \Omega}{[\Gamma, x' : \{x : X \mid \phi(x)\}]\ \vdash \phi(\mathsf{m}_\phi x')}$$

$$\frac{[\Gamma]\ \mathsf{the}\ x^X \mathsf{such\ that}\ \phi(x) : X}{[\Gamma]\ \vdash \phi(\mathsf{the}\ x^X \mathsf{such\ that}\ \phi(x))}$$

Rules related to equality

$$\frac{[\Gamma, \Gamma']\ \phi_i : \Omega \quad [\Gamma, \Gamma']\ \psi : \Omega \quad [\Gamma, \Gamma'[\vec{x} := \vec{s}]]\ \vec{\phi}[\vec{x} := \vec{s}] \vdash \psi[\vec{x} := \vec{s}] \quad [\Gamma]\ s_i = u_i : X}{[\Gamma, \Gamma'[\vec{x} := \vec{u}]]\ \vec{\phi}[\vec{x} := \vec{u}] \vdash \psi[\vec{x} := \vec{u}]}$$

$$\frac{[\Gamma]\ t = t' : X}{[\Gamma]\ \vdash \mathsf{Eq}(t, t')} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ t' : X \quad [\Gamma]\ \vdash \mathsf{Eq}(t, t')}{[\Gamma]\ t = t' : X}$$

Actually, we could have identified the judgement $[\Gamma]\ \vec{\phi} \vdash \psi$ with the equality judgement

$$[\Gamma]\ (\phi_1 \wedge \cdots \wedge \phi_n \to \psi) = \mathsf{true} : \Omega$$

and added extra equality rules; but we'll stick to our more familiar notation.

The other rules are essentially those of natural deduction. They are more convenient to reason with than the more symmetric Gentzen-style rules. For typographical reasons, they are presented in a 'linearised', sequent form. This means that we have to add exchange, weakening and contraction rules, to manage the lists of hypotheses.

Structural rules

$$\frac{[\Gamma]\ \vec{\phi}, \phi_0, \phi_1, \vec{\phi'} \vdash \psi}{[\Gamma]\ \vec{\phi}, \phi_1, \phi_0, \vec{\phi'} \vdash \psi} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi}{[\Gamma]\ \vec{\phi}, \phi' \vdash \psi} \qquad \frac{[\Gamma]\ \vec{\phi}, \phi', \phi' \vdash \psi}{[\Gamma]\ \vec{\phi}, \phi' \vdash \psi}$$

Logical rules

$$\frac{[\Gamma]\ \psi : \Omega}{[\Gamma]\ \psi \vdash \psi} \qquad \frac{}{[\Gamma]\ \ \vdash \mathsf{true}} \qquad \frac{[\Gamma]\ \psi : \Omega}{[\Gamma]\ \mathsf{false} \vdash \psi}$$

$$\frac{[\Gamma]\ \vec{\phi} \vdash \psi \quad [\Gamma]\ \vec{\phi} \vdash \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi \wedge \psi'} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi \wedge \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi \wedge \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi'}$$

$$\frac{[\Gamma]\ \vec{\phi} \vdash \psi}{[\Gamma]\ \vec{\phi} \vdash \psi \vee \psi'} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi \vee \psi'} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi \vee \psi' \quad [\Gamma]\ \vec{\phi}, \psi \vdash \zeta \quad [\Gamma]\ \vec{\phi}, \psi' \vdash \zeta}{[\Gamma]\ \vec{\phi} \vdash \zeta}$$

$$\frac{[\Gamma]\ \vec{\phi}, \psi \vdash \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi \rightarrow \psi'} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \psi \quad [\Gamma]\ \vec{\phi} \vdash \psi \rightarrow \psi'}{[\Gamma]\ \vec{\phi} \vdash \psi'}$$

$$\frac{[\Gamma, x : X]\ \vec{\phi(x)} \vdash \psi(x) \quad [\Gamma, x' : \{x : X \,|\, \psi(x)\}]\ \vec{\phi}(\mathsf{m}_\psi x') \vdash \psi'(x')}{[\Gamma, x : X]\ \vec{\phi(x)} \vdash \psi(x)\ \mathsf{andalso}\ \psi'(x)}$$

$$\frac{[\Gamma, x : X]\ \vec{\phi(x)} \vdash \psi(x)\ \mathsf{andalso}\ \psi'(x)}{[\Gamma, x : X]\ \vec{\phi(x)} \vdash \psi(x)} \qquad \frac{[\Gamma, x : X]\ \vec{\phi(x)} \vdash \psi(x)\ \mathsf{andalso}\ \psi'(x)}{[\Gamma, x' : \{x : X \,|\, \psi(x)\}]\ \vec{\phi}(\mathsf{m}_\psi x') \vdash \psi'(x')}$$

$$\frac{[\Gamma, x : X]\ \vec{\phi} \vdash \psi(x)}{[\Gamma]\ \vec{\phi} \vdash \forall x : X.\, \psi(x)} \text{ (provided } x \notin FV(\vec{\phi})\text{)} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \forall x : X.\, \psi(x) \quad [\Gamma]\ t : X}{[\Gamma]\ \vec{\phi} \vdash \psi(t)}$$

$$\frac{[\Gamma]\ t : X \quad [\Gamma]\ \vec{\phi} \vdash \psi(t)}{[\Gamma]\ \vec{\phi} \vdash \exists x : X.\, \psi(x)} \qquad \frac{[\Gamma]\ \vec{\phi} \vdash \exists x : X.\, \psi(x) \quad [\Gamma, x : X]\ \vec{\phi}, \psi(x) \vdash \zeta}{[\Gamma]\ \vec{\phi} \vdash \zeta} \text{ (provided } x \notin FV(\zeta)\text{)}$$

5: Rules for arithmetic

Dealing with the natural numbers involves introducing new judgements of all four kinds. It seemed convenient to put the new rules in a section of their own.

The natural numbers, and related terms

$$\frac{}{[\Gamma]\ \mathbb{N}\ \mathrm{set}} \qquad \frac{}{[\Gamma]\ 0 : \mathbb{N}} \qquad \frac{}{[\Gamma]\ \mathsf{s} : \mathbb{N}^{\mathbb{N}}} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ f : X^X}{[\Gamma]\ \mathsf{it}(f, t) : X^{\mathbb{N}}}$$

Behaviour of the iterator

$$\frac{[\Gamma]\ t : X \quad [\Gamma]\ f : X^X}{[\Gamma]\ \mathsf{it}(f,t)(0) = t : X} \qquad \frac{[\Gamma]\ t : X \quad [\Gamma]\ f : X^X}{\Gamma, x : \mathbb{N} \vdash \mathsf{it}(f,t)(\mathsf{s}(x)) = f(\mathsf{it}(f,t)(x)) : X}$$

Peano postulates

$$\frac{}{[\Gamma, x : \mathbb{N}] \vdash \neg \mathsf{Eq}(\mathsf{s}(x), 0)} \qquad \frac{}{[\Gamma, x : \mathbb{N}, y : \mathbb{N}]\ \mathsf{Eq}(\mathsf{s}(x), \mathsf{s}(y)) \vdash \mathsf{Eq}(x, y)}$$

$$\frac{[\Gamma, x : \mathbb{N}]\ \phi(x) : \Omega \quad [\Gamma] \vdash \phi(0) \quad [\Gamma, x : \mathbb{N}]\ \phi(x) \vdash \phi(\mathsf{s}(x))}{[\Gamma, x : \mathbb{N}] \vdash \phi(x)}$$

II: Interpretation in a topos

Let $\mathcal{E}$ be a topos. We will show how to use $\mathcal{E}$ to model our language, by interpreting set-expressions and terms as objects and arrows in $\mathcal{E}$—actually, in slices of $\mathcal{E}$. The soundness of the equality rules and the logical rules is mostly left as an exercise; we'll only go through a couple of examples.

Since we only have judgements relative to environments, we first have to decide how to interpret environments. If Γ is an environment, it will define a topos $[\![\Gamma]\!]$; for example, if Γ is the *empty* environment,

$$[\![\Gamma]\!] = \mathcal{E}$$

The judgements are interpreted as follows:

1. A derivable judgement
$$[\Gamma]\ X \text{ set}$$
will define an object $[\![X]\!] \in [\![\Gamma]\!]$. We can then put
$$[\![\Gamma, x : X]\!] \ = \ [\![\Gamma]\!]/[\![X]\!]$$
That is, extending a context with a free variable corresponds to moving to a slice of the original topos. As explained in the text, this captures the notion of 'variation over X'.

2. A derivable judgement
$$[\Gamma]\ t : X$$
will define an arrow $[\![t]\!]: 1 \to [\![X]\!]$ in $[\![\Gamma]\!]$; i.e. an 'element' of $[\![X]\!]$.

3. A judgement
$$[\Gamma]\ t = t' : X$$
will be *valid* relative to $\mathcal{E}$ exactly when
$$[\![t]\!] = [\![t']\!]: 1 \to [\![X]\!]$$

4. A judgement
$$[\Gamma]\ \vec{\phi} \vdash \psi$$
will be *valid* relative to $\mathcal{E}$ exactly when
$$[\![\phi_1]\!] \wedge \cdots \wedge [\![\phi_n]\!] \leq [\![\psi]\!]$$
in the Heyting algebra
$$\mathrm{Hom}_{[\![\Gamma]\!]}(1, \Omega) \ \cong \ \mathrm{Sub}_{[\![\Gamma]\!]}(1)$$
In particular, a judgement
$$[\Gamma] \ \dashv \psi$$
is valid if $[\![\psi]\!] = \top: 1 \to \Omega$ in $[\![\Gamma]\!]$.

The weakening and 'substitution for a variable' rules deserve some comment. Suppose $[\Gamma]\ Y$ set, so that we have an object $[\![Y]\!]$ in $[\![\Gamma]\!]$; if also $[\Gamma]\ X$ set, then weakening tells us that

$$[\Gamma, x : X]\, Y \text{ set}$$

Therefore we need to find an object $[\![Y]\!]'$ in $[\![\Gamma, x : X]\!] = [\![\Gamma]\!]/[\![X]\!]$. But we have the logical functor

$$[\![X]\!]^*\colon [\![\Gamma]\!] \to [\![\Gamma]\!]/[\![X]\!]$$

so we can just put $[\![Y]\!]' = [\![X]\!]^*[\![Y]\!]$.

Substitution for a variable is treated similarly. For example, consider the simple case

$$\frac{[\Gamma, x : X]\, Y \text{ set} \quad [\Gamma]\, t : X}{[\Gamma]\, Y[x := t] \text{ set}}$$

The hypotheses give us an object $[\![Y]\!]$ in $[\![\Gamma, x : X]\!] = [\![\Gamma]\!]/[\![X]\!]$ and an arrow $[\![t]\!]\colon 1 \to [\![X]\!]$ in $[\![\Gamma]\!]$. But this arrow induces the logical functor

$$[\![t]\!]^*\colon [\![\Gamma]\!]/[\![X]\!] \to [\![\Gamma]\!]$$

so we can just put $[\![Y[x := t]]\!] = [\![t]\!]^*[\![Y]\!]$.

Interpretation of sets

This is straightforward. Cartesian products are interpreted by (categorical) products, disjoint unions by binary coproducts and function spaces by exponentials. That is, we put

$$[\![1]\!] = 1 \qquad [\![\Omega]\!] = \Omega \qquad [\![\mathbb{N}]\!] = N$$

$$[\![X \times Y]\!] = [\![X]\!] \times [\![Y]\!] \qquad [\![X + Y]\!] = [\![X]\!] + [\![Y]\!] \qquad [\![Y^X]\!] = [\![Y]\!]^{[\![X]\!]}$$

all in the topos $[\![\Gamma]\!]$ corresponding to the environment Γ appearing in the judgement.

Subsets formed by separation If we've derived a judgement

$$[\Gamma, x : X]\ \phi(x) : \Omega$$

then we have an arrow $[\![\phi]\!]\colon 1 \to \Omega$ in $[\![\Gamma, x : X]\!] = [\![\Gamma]\!]/[\![X]\!]$; but this corresponds to an arrow

$$\Sigma_{[\![X]\!]}[\![\phi]\!]\colon [\![X]\!] \to \Omega$$

in $[\![\Gamma]\!]$. We define $[\![\{x : X \,|\, \phi(x)\}$ by an equalizer

$$[\![\{x : X \,|\, \phi(x)\}]\!] \xrightarrow{m_\phi} [\![X]\!] \underset{\top}{\overset{\Sigma_{[\![X]\!]}[\![\phi]\!]}{\rightrightarrows}} \Omega$$

in $[\![\Gamma]\!]$. Note that we have given a name to the mono $[\![\{x : X \,|\, \phi(x)\}]\!] \rightarrowtail [\![X]\!]$. For notational convenience, we often write

$$\|\phi\| = [\![\{x : X \,|\, \phi(x)\}]\!] \rightarrowtail [\![X]\!]$$

Indexed products and coproducts If we've derived a judgement

$$[\Gamma, x : X]\, Y(x) \text{ set}$$

then we have an object

$$[\![Y]\!] \in [\![\Gamma, x : X]\!] = [\![\Gamma]\!]/[\![X]\!]$$

But the functor $[\![X]\!]^*\colon [\![\Gamma]\!] \to [\![\Gamma]\!]/[\![X]\!]$ has right and left adjoints $\Pi_{[\![X]\!]}, \Sigma_{[\![X]\!]}\colon [\![\Gamma]\!]/[\![X]\!] \to [\![\Gamma]\!]$; so we let

$$[\![\prod_{x:X} Y(x)]\!] = \Pi_{[\![X]\!]}[\![Y]\!] \qquad [\![\coprod_{x:X} Y(x)]\!] = \Sigma_{[\![X]\!]}[\![Y]\!]$$

Interpretation of terms

Free variables To interpret the judgement

$$[\Gamma, x : X]\; x : X$$

we need an arrow $1 \to [\![X]\!]^*[\![X]\!]$ in $[\![\Gamma, x : X]\!] = [\![\Gamma]\!][\![X]\!]$; i.e. an arrow

$$[\![X]\!] \to [\![X]\!]$$

in $[\![\Gamma]\!]$. We just use the identity $1_{[\![X]\!]}$.

Elements of unit type and truth-values These are straightforward. For example, $[\![*]\!] = 1_1\colon 1 \to 1$, the identity map; and we interpret terms of type Ω using the internal Heyting algebra operations on Ω. For instance, if $[\Gamma]\;\phi : \Omega$ and $[\Gamma]\;\psi : \Omega$, $[\![\phi \wedge \psi]\!]\colon 1 \to \Omega$ is the composite

$$1 \xrightarrow{\langle [\![\phi]\!], [\![\psi]\!]\rangle} \Omega \times \Omega \xrightarrow{\quad\wedge\quad} \Omega$$

As for the predicates $\mathsf{Eq}(t, t')$: first note that for any object A in a topos, we have a diagonal map $\delta_A\colon A \rightarrowtail A \times A$, the 'equality relation' on A; let $eq_A\colon A \times A \to \Omega$ be its classifying map. Now suppose $[\Gamma]\; t : X$, $[\Gamma]\; t' : X$, so that we have maps $[\![t]\!], [\![t']\!]\colon 1 \to [\![X]\!]$ in $[\![\Gamma]\!]$. Then $[\![\mathsf{Eq}(t, t')]\!]\colon 1 \to \Omega$ is the composite

$$1 \xrightarrow{\langle [\![t]\!], [\![t']\!]\rangle} [\![X]\!] \times [\![X]\!] \xrightarrow{\quad eq_{[\![X]\!]}\quad} \Omega$$

We have to interpret $\phi(x)\,\mathsf{andalso}\,\psi(x)$. Suppose that $[\Gamma, x : \{x : X \,|\, \phi(x)\}]\;\psi(x) : \Omega$. We seek a suitable map $[\![X]\!] \to \Omega$. But we have the two monos

$$\|\psi\| \rightarrowtail \|\phi\| \rightarrowtail [\![X]\!]$$

The composite is mono, so we can let $[\![\phi(x)\,\mathsf{andalso}\,\psi(x)]\!]$ be its classifying map.

It remains to interpret the quantifiers. Suppose $[\Gamma, x : X]\;\phi(x) : \Omega$, giving us a map $[\![\phi(x)]\!]\colon 1 \to \Omega$ in $[\![\Gamma]\!]/[\![X]\!]$; this corresponds to a map $[\![X]\!] \to \Omega$ in $[\![\Gamma]\!]$, which by abuse of notation we call $[\![\phi]\!]$. We want to find maps

$$[\![\forall x : X.\, \phi(x)]\!], [\![\exists x : X.\, \phi(x)]\!]\colon 1 \to \Omega$$

in $[\![\Gamma]\!]$. Recall that $[\![\Gamma]\!](1, \Omega) \cong \mathrm{Sub}_{[\![\Gamma]\!]}(1)$, and $[\![\Gamma]\!]([\![X]\!], \Omega) \cong \mathrm{Sub}_{[\![\Gamma]\!]}([\![X]\!])$.

But $!_{[\![X]\!]}\colon [\![X]\!] \to 1$ induces a function $\mathrm{Sub}_{[\![\Gamma]\!]}(1) \to \mathrm{Sub}_{[\![\Gamma]\!]}([\![X]\!])$ with right and left adjoints

$$\forall_{!_{[\![X]\!]}}, \exists_{!_{[\![X]\!]}}\colon \mathrm{Sub}_{[\![\Gamma]\!]}([\![X]\!]) \to \mathrm{Sub}_{[\![\Gamma]\!]}(1)$$

So (using the above isomorphisms and abusing notation a bit more) we can let

$$[\![\forall x : X.\, \phi(x)]\!] = \forall_{!_{[\![X]\!]}}[\![\phi]\!], \quad [\![\exists x : X.\, \phi(x)]\!] = \exists_{!_{[\![X]\!]}}[\![\phi]\!]$$

Terms related to the natural numbers $[\![0]\!]$ and $[\![\mathsf{s}]\!]$ are modelled in the obvious way. For the iterator it, we use the universal property of the natural numbers object N. Suppose $[\Gamma]\ t : X$ and $[\Gamma]\ f : X^X$. Then we have an arrow $[\![t]\!]: 1 \to [\![X]\!]$ in $[\![\Gamma]\!]$; and the exponential transpose of $[\![f]\!]: 1 \to [\![X^X]\!] = [\![X]\!]^{[\![X]\!]}$ gives us an arrow $\overline{[\![f]\!]}: [\![X]\!] \to [\![X]\!]$ in $[\![\Gamma]\!]$. Then we have a diagram

$$\begin{tikzcd} & N \arrow[r, "s"] \arrow[d, dotted, "g"] & N \arrow[d, dotted, "g"] \\ 1 \arrow[ur, "0"] \arrow[dr, "{[\![t]\!]}"'] & & \\ & X \arrow[r, "\overline{[\![f]\!]}"'] & X \end{tikzcd}$$

$[\![\mathsf{it}(f,t)]\!]: 1 \to [\![X^{\mathbb{N}}]\!] = [\![X]\!]^N$ is the exponential transpose of the unique map g that makes the triangle and the square commute.

Cartesian products, disjoint unions and function spaces In each case we simply use the universal property of the object in question. We will describe the situation for disjoint unions and leave the rest to the reader.

If $[\Gamma]\ t : X$ then we have $[\![t]\!]: 1 \to [\![X]\!]$ in $[\![\Gamma]\!]$. We let $[\![\mathsf{in}_0 t]\!]: 1 \to [\![X+Y]\!]$ be the composite

$$1 \xrightarrow{[\![t]\!]} X \xrightarrow{\mathsf{in}_0} X+Y$$

in_1 is treated similarly.

Now suppose

$$[\Gamma]\ t : X+Y \qquad [\Gamma, x:X]\ u : Z \qquad [\Gamma, y:Y]\ u' : Z$$

Their interpretations give us maps

$$[\![t]\!]: 1 \to [\![X]\!]+[\![Y]\!] \qquad f = \Sigma_{[\![X]\!]}[\![u]\!]: [\![X]\!] \to [\![Z]\!] \qquad f' = \Sigma_{[\![Y]\!]}[\![u']\!]: [\![Y]\!] \to [\![Z]\!]$$

in $[\![\Gamma]\!]$. We let $[\![\mathsf{case}(t, x \triangleright u, y \triangleright v)]\!]: 1 \to Z$ be the composite

$$1 \xrightarrow{[\![t]\!]} [\![X]\!]+[\![Y]\!] \xrightarrow{[f,f']} [\![Z]\!]$$

Subsets formed by separation Here we use the fact that such subsets are modelled using equalizers. Suppose $[\Gamma]\ t : \{x \in X \,|\, \phi(x)\}$, so that $[\![t]\!]: 1 \to \|\phi\|$; then we define $[\![\mathsf{m}_\phi t]\!] = m_\phi[\![t]\!]$, where $m_\phi: \|\phi\| \rightarrowtail [\![X]\!]$ is the mono appearing in the equalizer diagram defining $\|\phi\|$.

Conversely, suppose $[\Gamma]\ t : X$ and $[\Gamma]\ \vdash \phi(t)$. Since the interpretation of logical judgements is sound, we must have $[\![\phi(t)]\!] = \top$; that is, $[\![t]\!]$ equalizes $\Sigma_{[\![X]\!]}[\![\phi]\!]$ and $\top$. So we let $[\![\mathsf{p}_\phi t]\!]: 1 \to \|\phi\|$

be the unique map making the following diagram commute:

$$\begin{tikzcd} \|\phi\| \arrow[r, "m_\phi"] & \llbracket X \rrbracket \arrow[r, shift left, "\Sigma_{\llbracket X \rrbracket}\llbracket \phi \rrbracket"] \arrow[r, shift right, "\top"'] & \Omega \\ 1 \arrow[u, dotted, "\llbracket \mathsf{p}_\phi t \rrbracket"] \arrow[ur, "\llbracket t \rrbracket"'] & & \end{tikzcd}$$

Indexed products and coproducts First we do indexed products. If $[\Gamma, x : X]\ t : Y(x)$ then we have a map

$$\llbracket t \rrbracket : \llbracket X \rrbracket^* 1 \cong 1 \to \llbracket Y \rrbracket$$

in $\llbracket \Gamma \rrbracket / \llbracket X \rrbracket$. The desired map

$$\llbracket \langle x \rangle t \rrbracket : 1 \to \Pi_{\llbracket X \rrbracket} \llbracket Y \rrbracket$$

in $\llbracket \Gamma \rrbracket$ is its transpose under the adjunction $\llbracket X \rrbracket^* \dashv \Pi_{\llbracket X \rrbracket}$.

Now suppose

$$[\Gamma]\ t : \prod_{x:X} Y(x) \qquad\qquad [\Gamma]\ s : X$$

so that we have maps

$$\llbracket t \rrbracket : 1 \to \Pi_{\llbracket X \rrbracket} \llbracket Y \rrbracket \qquad\qquad \llbracket s \rrbracket : 1 \to \llbracket X \rrbracket$$

in $\llbracket \Gamma \rrbracket$. Then if we let

$$\overline{\llbracket t \rrbracket} : \llbracket X \rrbracket^* 1 \cong 1 \to \llbracket Y \rrbracket$$

be the transpose of $\llbracket t \rrbracket$ under the adjunction $\llbracket X \rrbracket^* \dashv \Pi_{\llbracket X \rrbracket}$, we can let

$$\llbracket \pi_s t \rrbracket = \llbracket s \rrbracket^* \overline{\llbracket t \rrbracket} : 1 \to \llbracket Y(s) \rrbracket = \llbracket s \rrbracket^* \llbracket Y \rrbracket$$

What about indexed coproducts? If

$$[\Gamma]\ s : X \qquad\qquad [\Gamma]\ t : Y(s)$$

so that we have maps

$$\llbracket s \rrbracket : 1 \to \llbracket X \rrbracket \qquad\qquad \llbracket t \rrbracket : 1 \to \llbracket s \rrbracket^* \llbracket Y \rrbracket$$

in $\llbracket \Gamma \rrbracket$. Recall that if $\llbracket Y \rrbracket = \left(\begin{smallmatrix} Y \\ \downarrow f \\ \llbracket X \rrbracket \end{smallmatrix} \right) \in \llbracket \Gamma \rrbracket / X$, then

$$\Sigma_{\llbracket X \rrbracket} \llbracket Y \rrbracket = Y \in \mathcal{E}$$

and $\llbracket s \rrbracket^* \llbracket Y \rrbracket$ is calculated as a pullback

$$\begin{tikzcd} \llbracket s \rrbracket^* \llbracket Y \rrbracket \arrow[r, "\mathsf{in}_s"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "f"] \\ 1 \arrow[r, "\llbracket s \rrbracket"'] & \llbracket X \rrbracket \end{tikzcd}$$

Then $[\![\mathsf{in}_s t]\!]: 1 \to \Sigma_{[\![X]\!]}[\![Y]\!]$ is the composite

$$1 \xrightarrow{[\![t]\!]} [\![s]\!]^*[\![Y]\!] \xrightarrow{\mathsf{in}_s} Y$$

Conversely, suppose $[\Gamma]\ t : \coprod_{x:X} Y(x)$, so that

$$[\![t]\!]: 1 \to \Sigma_{[\![X]\!]}[\![Y]\!] = Y$$

Then $[\![\mathsf{which}(t)]\!]: 1 \to [\![X]\!]$ is the composite

$$1 \xrightarrow{[\![t]\!]} Y \xrightarrow{f} [\![X]\!]$$

To compute

$$[\![\mathsf{what}(t)]\!]: 1 \to [\![\mathsf{which}(t)]\!]^*[\![Y]\!]$$

first note that $[\![\mathsf{which}(t)]\!]^*[\![Y]\!]$ is computed by pulling back Y as above (put $s = [\![\mathsf{which}(t)]\!]$); then note that because the square is a pullback, $[\![t]\!]$ must factorize through $\mathsf{in}_{[\![\mathsf{which}(t)]\!]}$. This gives us $[\![\mathsf{what}(t)]\!]$.

Definite descriptions This is the only case that takes a little work; the interpretation is due to Kock. We give a rather categorical account of it.

Suppose that

$$[\Gamma]\ \vdash \exists x : X.\, \phi(x)$$

This implies that the image of the map $\|\phi\| \overset{m_\phi}{\rightarrowtail} X \to 1$ is 1. Now the image of the map $\|\phi\| \to 1$ is the coequalizer of its kernel pair, which is just $\pi_0, \pi_1 : \|\phi\| \times \|\phi\| \to \|\phi\|$; in other words, we have a coequalizer diagram

$$\begin{tikzcd} \|\phi\| \times \|\phi\| \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] & \|\phi\| \arrow[rr] & & 1 \end{tikzcd}$$

Now suppose that

$$[\Gamma]\ \vdash \forall x, x' : X.\, \phi(x) \wedge \phi(x') \to \mathsf{Eq}(x, x')$$

This implies that $[\![\phi(x)]\!] \wedge [\![\phi(x')]\!] \leq [\![\mathsf{Eq}(x, x')]\!]$ in $\mathrm{Sub}_{[\![\Gamma]\!]}(X \times X)$. So there is a fill-in h in the diagram

$$\begin{tikzcd} \|\phi\| \times \|\phi\| \arrow[rr, dotted, "h"] \arrow[dr, "m_\phi \times m_\phi"'] & & X \arrow[dl, "\delta_X"] \\ & X \times X & \end{tikzcd}$$

Also, the square

$$\begin{tikzcd} \|\phi\| \times \|\phi\| \arrow[r, "m_\phi \times m_\phi"] \arrow[d, shift right, "\pi_0"'] \arrow[d, shift left, "\pi_1"] & X \times X \arrow[d, shift right, "\pi_0"'] \arrow[d, shift left, "\pi_1"] \\ \|\phi\| \arrow[r, "m_\phi"'] & X \end{tikzcd}$$

clearly commutes (serially; i.e. the obvious two squares commute). We have abused notation slightly by giving the projections maps the same names.

It follows that $m_\phi \pi_0 = m_\phi \pi_1$. For in the diagram

$$\begin{tikzcd}
 & X \arrow[d, "\delta_X"'] \arrow[dr, "1_X"] & \\
\|\phi\| \times \|\phi\| \arrow[ur, "h"] \arrow[r, "m_\phi \times m_\phi"] \arrow[dr, shift right, "\pi_0"'] \arrow[dr, shift left, "\pi_1"] & X \times X \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] & X \\
 & \|\phi\| \arrow[ur, "m_\phi"'] &
\end{tikzcd}$$

we have

$$m_\phi \pi_0 = \pi_0(m_\phi \times m_\phi) = \pi_0 \delta_X h = 1_X h = \pi_1 \delta_X h = \pi_1(m_\phi \times m_\phi) = m_\phi \pi_1$$

Hence we can let $[\![\textsf{the } x^X \textsf{such that } \phi(x)]\!] : 1 \to [\![X]\!]$ be the unique arrow such that the following diagram commutes:

$$\begin{tikzcd}
\|\phi\| \times \|\phi\| \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] & \|\phi\| \arrow[r] \arrow[dr, "m_\phi"'] & 1 \arrow[d, dotted, "{[\![\textsf{the } x^X \textsf{such that } \phi(x)]\!]}"] \\
 & & {[\![X]\!]}
\end{tikzcd}$$

Actually, there is more than one way of interpreting definite descriptions, though of course they all give the same answer. It is also possible to interpret indefinite descriptions, though we will not discuss those.

Soundness of equality and logical judgements, and arithmetic rules

The rule

$$x = x : X$$

is trivially sound. So is the 'substitute equals for equals' rule, because making the substitution

$$\vec{x} := \vec{s}$$

corresponds to applying a logical functor $[\![\vec{s}]\!]^*$; and since $[\![s_i]\!] = [\![t_i]\!]$,

$$[\![\vec{s}]\!]^* = [\![\vec{t}]\!]^*$$

The rest of the equality rules follow from the universal properties of the objects used to interpret sets.

The basic logical rules are trivially sound.

The soundness of substituting for a variable in a logical judgement follows from the fact that pullback functors are logical, and thus preserve entailment in Sub(1) as well as the interpretations of the logical connectives.

The rules relating Eq and $=$ are sound because $eq_{[\![X]\!]}$ classifies the equality relation on $[\![X]\!]$. The fact that the structural logical rules and the rules for the propositional connectives are sound follows from the fact that the Heyting algebra Sub(1) provides a sound interpretation of constructive propositional logic. The soundness of the rules for $\mathsf{andalso}$ and for the quantifiers is also easy to check.

The arithmetic rules are also straighforward, except for the soundness of the induction schema, which is quite difficult to verify: see the references.

The internal language of a topos

So far we've only been interpreting the 'bare' language inside our topos $\mathcal{E}$. It may not be expressive enough to talk about all the aspects of $\mathcal{E}$ that we're interested in; for example, there may be interesting objects and arrows in $\mathcal{E}$ that are not definable in the bare language. So it's convenient to be able to extend the language with new expressions for sets and with new terms, to describe these objects and arrows.

If A is an object of $\mathcal{E}$, we can add a new symbol A to our language, together with a new judgement

$$\frac{}{[\,]\ \mathsf{A}\ \mathrm{set}}$$

Then we simply define $[\![\mathsf{A}]\!] = A$ in $\mathcal{E}$. Actually, we often abuse notation and write "A" instead of A.

If $f\colon A \to B$ is an arrow in $\mathcal{E}$, and we have added new symbols A, B to the language to stand for A and B, we can add a new symbol f together with a judgement

$$\frac{}{[\,]\ \mathsf{f} : \mathsf{B}^{\mathsf{A}}}$$

Of course, there will usually be plenty of equality judgements and logical judgements valid relative to $\mathcal{E}$ which are not provable. If

$$[\Gamma]\ \cdots$$

is such a judgement, then we add it to our list of basic judgements. That is, once we know a judgement to be valid, we are allowed to use it to derive new judgements.

If we do this for *all* the objects and arrows of $\mathcal{E}$, we obtain the **internal language** of $\mathcal{E}$. It can be shown that $\mathcal{E}$ and its internal language essentially determine each other: see, for example, Lambek/Scott[15].

It's quite a lot of work to make all this completely precise and check that it makes sense. That task is beyond the scope of these notes.

[15]Since we have definite descriptions and subsets defined by separation, there is in our treatment a rather closer relationship between the languages and the toposes.

III: Rough guide to the effective topos

The following figures summarise the relationships between the different categories and fibrations defined in chapter 4. $\mathcal{E}ff$ has various other interesting full subcategories; however, these were not discussed in the text and therefore do not appear in these pictures.

$$
\begin{tikzcd}
\begin{array}{c} \mathcal{S}ets \simeq \\ (\neg\neg\text{-sheaves in } \mathcal{E}ff) \end{array} \arrow[dr] & & \\
& \begin{array}{c} \omega\text{-}\mathcal{S}ets \simeq \\ (\neg\neg\text{-separated objects in } \mathcal{E}ff) \end{array} \arrow[r] & \mathcal{E}ff \\
\begin{array}{c} \mathsf{Per} \simeq \\ (\text{effective } \omega\text{-sets}) \simeq \\ (\text{modest sets in } \mathcal{E}ff) \end{array} \arrow[ur] & &
\end{tikzcd}
$$

all the arrows denote full reflective embeddings

Figure 4.1: Relationships between categories, simplified version

$$
\begin{tikzcd}
\begin{array}{c} \mathsf{Per}^{\rightarrow} \\ \downarrow \\ \mathsf{Per} \end{array} \arrow[dr] & & \begin{array}{c} \omega\text{-}\mathcal{S}ets^{\rightarrow} \\ \downarrow \\ \omega\text{-}\mathcal{S}ets \end{array} \arrow[dr] & & \\
& \begin{array}{c} \mathbb{P} \\ \downarrow \\ \omega\text{-}\mathcal{S}ets \end{array} \arrow[ur, dotted] \arrow[dr] & & \begin{array}{c} \mathcal{S}ep \\ \downarrow \\ \mathcal{E}ff \end{array} \arrow[r, dotted] & \begin{array}{c} \mathcal{E}ff^{\rightarrow} \\ \downarrow \\ \mathcal{E}ff \end{array} \\
\begin{array}{c} \mathbb{P} \\ \downarrow \\ \mathcal{S}ets \end{array} \arrow[ur] & & \begin{array}{c} \mathcal{M}od \\ \downarrow \\ \mathcal{E}ff \end{array} \arrow[ur, dotted] & &
\end{tikzcd}
$$

dotted arrows denote full reflective subfibrations
solid arrows denote fibrations obtained by change of base

Figure 4.2: Relationships between fibrations, slightly simplified version

Index