Recursive Domains, Indexed Category Theory and Polymorphism

Paul Taylor

1983–7

A dissertation submitted
for the degree of
Doctor of Philosophy
at the
University of Cambridge

also

for the Summer 1986 Prize Fellowship Competition at
Trinity College, Cambridge

I hereby declare that my thesis entitled
Recursive Domains, Indexed Category Theory and Polymorphism
is not substantially the same as any that I have submitted
for a degree or diploma or other qualification at any other University.
I further state that no part of thesis has already been (or is being)
concurrently submitted for any such degree, diploma or other qualification.

Further,

I declare that the results presented herein
are the product of my own original research
where claimed as such.
This qualification will be explained in the Introduction.

During the course of my research I have been supported financially by
the Science and Engineering Research Council,
Trinity College
and the Department of Pure Mathematics and Mathematical Statistics.

This dissertation (in particular Chapter V) has been revised somewhat since submission.

This work is dedicated with love to
the guardians of my sanity

James Annett
Mark Dawson
Ellie Mayger (*née* Clarke)
Mike Mayger
Fiona Miller
Shane Voss
Ian White

without whose support
it would never have happened.

Introduction

Let me begin by confessing that this is a very eccentric document. It aims to do two things. First, of course, it sets out some original constructions, primarily of a "type of types" in certain categories whose objects have come to be known (rather unimaginatively) as domains, but also of various other things in such categories. Second, however, which is a departure from custom as far as Ph.D. or fellowship dissertations are concerned, it attempts to give an introduction to a substantial but largely undocumented body of knowledge among (certain) Mathematicians, namely Categorical Logic, which has very recently been found to be of considerable importance in Theoretical Computer Science.

Category Theory, I like to say, is a *tourist subject*. It started in Group Theory, but was only taken seriously when Frank Adams needed it to discuss constructions in Topology; after that it visited Linear Algebra, Universal Algebra, Automata Theory and Algebraic Geometry. At the last of these it met Logic and turned into Topos Theory. The journey to that stage was undertaken (almost) entirely in the company of mathematicians, who could perhaps be expected to understand the significance of the *souvenirs*, and in particular the geometrical language. Now that it has arrived in Computer Science, however, I feel that some basic discussion of what it's all about is called for. This is what I have been trying to do.

I have a strong conviction that when one has begun to understand something which is understood by few other people then it is one's duty to formulate on paper that understanding (in whatever terms one has found useful) for the benefit of others trying to do likewise. Moreover I have an habitual urge to do so. This is my defence for the length and verbosity of this work.

This policy also requires me to apologise to the originators of the results, and other practitioners of the subject far more expert than myself, for tiring them with the necessity of reading my misrepresentations. This I freely do.

I have also included a lot of standard Domain Theory, starting with the Lambda Calculus. **This is not a thesis about the Lambda Calculus!** This is because I disagreed with some of the existing terminology and because I wished to push the results as far as I could. Also, in Chapter V I wanted to be able to assume that a category of domains was of the form $\mathbf{Retr}(\Lambda)$; this explains my preference for continuous rather than algebraic domains.

Since much of the bulk of the work is therefore "standard" material, a notation is called for to indicate what is *original*. This is done by means of a "+" before the number of the section. The other sections I claim for the most part to be an original *account* of their subject, though occasionally I have marked sections "−" by way of confession of having forgotten the proof of the result in question. In some cases originality means improvement (or in a few cases rediscovery) of an existing result; here credit has been given in apparent contradiction to the mark on the section.

Certain results in this work have been designated as **Fact**s. (Specifically in §§1.1.13, 1.2.11, 2.2.11, 2.5.8.) This indicates an original result which I have not had time to write up.

The most substantial of these is a technique for dealing with large filtered diagrams in a small category (where we just say "cardinality" in a poset), the proof of which would have taken ten pages. [POSTSCRIPT: This result is a factorisation system which is similar to that of any functor into a final functor followed by a discrete fibration [Street & Walters, 1973?]. The difference is that the first part of my factorisation is also regular epi whilst the second is characterised by a condition I called *cosignposted*; the latter places a bound on the size of the intermediate category in terms of that of the codomain, whilst the former retains the property of final functors that they yield equivalent filtered colimits. The final-fibration factorisation for functors which preserve connected limits is the same as that giving Diers' spectrum and Berry's trace.]

I also have another result whose proof is of a similar length but which is not mentioned in the text. This is that an interpretation by means of partial maps can be given to s simple imperative language (including a **while** construct) in a category with certain *finitary* exactness conditions. Specifically, we use equalisers of a form in which one of the maps is a mono to interpret the **while**; the important stage is the proof (in an elementary topos, though the category itself does not need higher order logic) that a certain graph has the diamond property and hence that the map from the equaliser to the coequaliser of a pair of maps one of which is mono is mono. There is no mention of anything of an infinitary nature in this proof: it uses the adjoint functor theorem in the way in which §4.4 tries to indicate that it can be used to replace infinitary arguments.

The history of Computer Science from Babbage onwards is one of *generalisation* and *abstraction*. Foremost among the ideas are those of subroutines and high-level (*i.e.* machine-independent) languages, enabling the essence (algorithm) of some process to be expressed without reference to either its application or implementation. As programs become ever more complicated, it becomes necessary to provide *formal* rather than *ad hoc* techniques for understanding them and proving them correct. This is the rôle of Mathematics, specifically Category Theory, in theoretical Computer Science.

The major languages which were available for programming until the end of the 1960s were *imperative*, which is to say they set out precisely how — and in particular in what order — the operations were to be performed. The exception to this was LISP. This was nominally based on the λ-calculus, a mathematical tool for the study of recursion with its origins in symbolic logic. LISP and modern theoretical languages are called *functional* because they express only the behaviour of the program as a function from inputs to outputs, abstracting away details of sequencing just as earlier high level languages had hidden space allocation. This change shifts the emphasis as regards the *meaning* of the program from the *operational* to the *mathematical* viewpoint. It also places functions on the same footing as data items. Also, whereas the imperative style provided recursion (inductive definition) by means of program loops or jumps, the functional style expresses the function as a solution of an equation. This is always of a specific form, a *fixed point*, so for instance the factorial function is the fixed point of

$$f \quad \mapsto \quad \left(n \quad \mapsto \quad \begin{cases} 1 & \text{if } n = 0 \\ n \times f(n-1) & \text{otherwise} \end{cases} \right)$$

which is a function $\mathbb{N} \to \mathbb{N}$. Since we have placed no restriction on the nature of this function (and cannot, because of the *Halting Problem*), the domains on which it is defined cannot be ordinary sets.

Another feature of some of the modern languages is *polymorphism*, which is the abstraction of the function away from the types of its arguments; for instance the same code may be used to calculate determinants of matrices with either real or complex entries. Stronger forms of polymorphism may also be contemplated, as far as asking for a *type of all types*.

It is clear that concepts such as these cannot be expressed straightforwardly in terms of discrete sets. We can express them in the same way as is customary in existing imperative programming practice, namely operationally. However to do this rigorously we more or less have to invent or describe a particular machine, and this immediately loses the abstraction which we have tried to gain.

In the absence of recursion there are mathematical interpretations of functional languages and even polymorphism naturally provided by category theory. The former is clearly described by a *cartesian closed category*, which has an exponential or function space operation. The determinant example above, which is *uniform* over the ring definition, is reminiscent of the naturality of the double dual in Linear Algebra which led Eilenberg and Mac Lane to category theory. In fact this example can be used to motivate the idea of a classifying topos. Once we include recursion this approach breaks down; the λ-calculus, for instance, had no naturally occurring model before Scott replaced *function* by *continuous function* and built the so-called D_∞ and $P\omega$ models. This led to the notion of a *domain*, which is a particular kind of space in which recursion by means of fixed points is possible. $P\omega$, also provides an *ad hoc* model for recursion. Following this there was considerable work in the theoretical computer science community on extending the Scott-Strachey programme to nondeterministic and parallel languages.

The major aim of the present work is the discussion of polymorphism. This idea arises in a spectrum from the "uniform constructions" such as the determinant which may be expressed in terms of categorical techniques motivated essentially by the category of sets, to the most extreme form in which we admit the type of all types as a first class type itself. Of course this approach has frequently led to contradiction in the past.

We begin by establishing the major standard results in the theory of domains. The first chapter looks at the λ-calculus, the category of retracts, the $P\omega$ model and various notions of fixed point.

Chapter II moves to categories of domains and solutions of recursive domain equations. Having provided a *definition* of a category of domains, it shows that any such arises as a category of retracts.

In chapter III the various forms of polymorphism are discussed. We find that *indexed category theory* is the appropriate tool for discussing them all. A notion of "quantification" over types (where we consider the identity not to be a function which exists separately once for each type, but has a type "$\forall X.X \to X$") is found to correspond to completeness of the category in the sense of having all "small" products.

Chapter IV reformulates *indexed* as *fibred* categories, presenting a body of standard theory not currently well represented in the literature. This was motivated by the category of sets, but the present treatment makes certain generalisations, in particular by removing the requirement for *all* pullbacks (since categories of domains do not possess them) to give the notion of a *relatively cartesian closed category*.

In chapter V original material finally forms the major part, where the foregoing theory is applied to categories of domains. Whereas the only model with a type of types currently to be found in the literature is that based on $P\omega$, we find that *any* category of domains in fact possesses one. Moreover the crudity of the manner of construction shows that models of this nature abound.

Many previous formulations of universal sets and types of types have fallen to contradiction. We present a new paradox along these lines, which essentially shows that the three notions of *equality*, *function spaces* and *type of types* (at least as they are understood in category theory) cannot coexist. The model constructed in chapter V fails to express equality, and indeed most of the traditional notions in logic (comprehension, conjunction, disjunction and negation) are also necessarily absent from it.

As the main conceptual contribution of this work, I offer the notion of a *continuous type-dependence*. Of course we have to pluck something out of the air as a definition for this, but I believe that certain pieces of evidence strongly justify my choice.

First, it is naturally suggested by examination of the obvious idea with the category of retracts (§5.1.13).

Second (§5.2.2), it proves to be the only thing which works (contrary to the claim of my Surrey paper [1986]).

Third, it also works in arbitrary categories of boundedly complete domains (§5.2.9). Though we have a better grasp on these than on arbitrary posets, there is likely to be quite a rich structure in the hierarchy and our ability to deal with fibrations of them seems a remarkable coincidence. Conversely this partly justifies my rather perverse maintainance of such a wide range of categories of domains; on the other hand there is this apparent correspondence with fragments of logic and perhaps there is some "semantic complexity" notion to be found.

I should like to make a few personal comments on the subject.

It will be apparent that I consider myself a Mathematician rather than a Logician (I shall avoid the question as to whether I am a Computer Scientist), in the sense that it is tangible mathematical objects (models) which interest me and not the manipulation of syntax. In fact syntactic discussion of polymorphic languages is decidedly thin in this work. As regards the "denotational semantics" of programming languages, I lost (or rather, suspended) my faith in the applicability of these techniques some two years ago; my interest in the subject is in the remarkable *mathematical* behaviour of categories of domains.

I have perhaps been less than courteous to some of its other practitioners. I apologise for this. However it would appear that Scott's original $P\omega$ and D_∞ constructions [1972] and his *Data Types as Lattices* paper [1976], together with the work of Smyth and Plotkin on recursive domain equations [1978] account for most of what has been done as far as semantics, *i.e.* the provision of mathematical objects in which to explain things, is concerned. I am of the belief that Chapter V of the present work is the first model of a type-of-types since 1976. Moreover I further claim that it is the first (ever) to attempt to explain its own existence (I cannot help but regard V in $P\omega$ as an accident). This in no way detracts from the value of the [1976] paper, indeed it is a veritable goldmine containing many nuggets of which I personally have very little understanding and which are sure to lead to many further major contributions to the subject.

I should also like to offer a piece of friendly advice to anyone who might attempt in the future to study Domain Theory. It is this.

> *Does your conjecture make any nontrivial statement about finite posets?*
> *If so, then it's false: look for the counterexample.*

On the other hand the remarkable fact about categories of domains is that we seem never to have to worry about questions of size; indeed this (the fact that the function-space of a continuous lattice is essentially no bigger than the given lattice) is what led Scott into the subject in the first place. In other words, the existence of a type-of-types is no particular surprise.

Finally I should like to acknowledge those who have assisted and advised me during the past three years. Foremost among these is *Martin Hyland*, whose badgering led to my thinking and subsequently writing about Polymorphism in the first place. Many ideas of his have found their way into this text, and though the style of chapter III is not what one usually expects in Mathematics (and in no way is he to be blamed for it), this is where most of them have come to rest. Chapter IV is the result of the influence of my supervisor, *Peter Johnstone*; indeed it is an account of his Michaelmas 1982 Part III course). Then there are (with a little licence) the members of the joint DPMMS and Computer Laboratory seminar, *Mike Abbott*, *Jon Fairbairn*, *Thomas Forster*, *Carl Gunter*, *Alan Mycroft*, *Larry Paulson*, *Edmund Robinson*, *Giuseppe Rosolini*, *Glynn Winskel* and others. Outside Cambridge, I have had most enlightening discussions with *Samson Abramsky*, *Mike Fourman*, *Eugenio Moggi*, *Andrew Pitts*, *Mike Smyth*, *Steve Vickers* and *Gavin Wraith*. All these people I should like to thank for their patience with my troublesome manner and eccentric approach. Finally *Chris Thompson* of the Computing Service has provided more help with "TEX" than I have had time to use.

POSTSCRIPT, FEBRUARY 1987: The dissertation as presented here has been revised somewhat since its original submission. The main change is that most of the last chapter has been rewritten, and in particular the proofs have been reworked. This is the fruit of my attempting to explain the work to my new colleagues at Imperial College (in particular Samson Abramsky, *Yves Lafont*, *Luke Ong*, *Axel Poigné*, Mike Smyth and Steve Vickers). During this time I have also had the pleasure of meeting *Gordon Plotkin* for the first time, and there are a few comments in §2.5 resulting from this. This document has behaved rather like a large piece of software, always in need of maintainance and departing rapidly further away from perfection. I should like to apologise to all those to whom I promissed copies six months ago for succumbing to the temptation to keep revising it, and to those who read it now for the many residual errors. Finally I should like to record my appreciation to my examiners, Gavin Wraith and Glynn Winskel for their patience in reading (and approving) my thesis.

MARCH 1991: Despite its many errors and idiosyncracies, there still remains demand for copies of this dissertation. I would like to apologise to those who asked for copies of it as long as three years ago. The reason why it has been "out of print" is that it was originally written using a bug-ridden wysiwyg word processor and translated into "`plain` TEX"; the latter relied upon a collection of *ad hoc* additional macros, which after a while ceased to work. It has now been translated into LATEX, and as a result has got its index back, though the bibliography is rather inaccurate and out of date. All of the diagrams (some of which were literally "scissors and paste" inclusions) have been redrawn. To any reader who is thinking of writing a book using some wysiwyg word-processor and translating it, I have a simple word of advice: don't.

During this intervening time, of course much has happened in domain theory and polymorphism. A whole new kind of domain theory, characterised by *wide pullbacks* in addition to the filtered colimits which characterise the kind discussed herein, has seen its rise, fall and renaissance. My own interest now lies in the *synthetic* kind of which the *Effective Topos* is the most famous model.

The axiomatisation of categorical models of polymorphism introduced here (*relatively cartesian closed categories*) has become standard, and has been developed by several authors, notably Thomas Ehrhard and Thomas Streicher. The problem of the largest cartesian closed category of algebraic IPOs has also been solved by *Achim Jung,* whom I would also like to add to the foregoing list of acknowledgements (both as a colleague and as a friend).

JANUARY 1992: It is inappropriate to make alterations to a thesis, or to any document which has been overtaken by history or changes in its author's style. However before catching up with the requests for copies mentioned above and making the text available by FTP

`theory.doc.ic.ac.uk` `/theory/papers/Taylor/thesis`

I have decided to remove a number of fallacies. These have been replaced by counterexamples or references where available, and I am particularly grateful to Achim Jung for his assistance with many of these corrections. In the interests of historical accuracy I have left markers admitting to my mistakes and have resisted the temptation to give any generalations or anachronistic references. In several cases sketchy proofs here are filled in in my unpublished 1987 paper, *Homomorphisms, bilimits and saturated domains*, which is also available by FTP as `bilimits`.

Contents

Chapter 1

Categories with Models of the Lambda Calculus

1.1 Lambda Calculus and Combinatory Algebra

§**1.1.1** This chapter is largely concerned with Church's Lambda Calculus, and how it may be used to obtain a category of domains; the next chapter discusses the latter in the abstract. The λ-calculus is to be regarded in this work as a *vehicle* for other study rather than the prime object of interest. Though most of the material is standard, we wish to establish notations and conventions and make appropriate remarks concerning its relevance to later parts of the work.

The λ-calculus is concerned with the behaviour of functions under *application* rather than *composition* as in category theory. It comes in two forms, typed and untyped. The *typed* λ-calculus is simply a syntax for cartesian closed categories. A model of the *untyped* λ-calculus would be a set Λ whose function space Λ^Λ is a retract of Λ; though this is clearly impossible for discrete sets by cardinality, the purpose of the first three sections is to justify this as a definition. Here every function has a fixed point, which enables us to interpret *recursion*, indeed there is a uniform way of assigning such fixed points.

We have already stumbled upon the major concepts to be discussed in this chapter (indeed most of those with which the work is concerned), namely *retracts*, *cartesian closed categories*, *fixed points* and *typing*. We shall also discuss a popular naturally occurring model, known as the $P\omega$ model.

For a comprehensive treatment of the λ-calculus see Barendregt [1981], Koymans [1984] and Lambek & Scott [1986]. Basic category theory will be assumed without comment: see Mac Lane [1971] or Arbib and Manes [1975].

§**1.1.2** Here is the definition in a "telegraphic" form; for more detail see Barendregt §2.1. P_f denotes the *finite powerset*, *i.e.* the set of finite subsets of a set.

Suppose we have an arbitrary collection Σ of symbols (variables), along with the ability to generate a (countable) potential infinity of additional ones. We shall often write $\vec{x}$ for a set of variables, usually intending it to be the string $x_1, x_2, ..., x_n$.

By a simultaneous recursion we define the set $\mathsf{\Lambda}^+[\Sigma]$ of *raw lambda terms* in the variables Σ, the function $\mathsf{FV} : \mathsf{\Lambda}^+[\Sigma] \to P_f(\Sigma)$ assigning to a term its set of *free variables* and the function $(-)[- := -] : \mathsf{\Lambda}^+[\Sigma] \times \Sigma \times \mathsf{\Lambda}^+[\Sigma] \to \mathsf{\Lambda}^+[\Sigma]$ which is the *substitution* in one term, for all free occurrences of a certain variable, another term.

if...	then... $\in \mathsf{\Lambda}^+[\Sigma]$	with $\mathsf{FV} = ...$	and $-[y := c] = ...$
$x \in \Sigma, x = y$	x	$\{x\}$	c
$x \in \Sigma, x \neq y$	x	$\{x\}$	x
$a, b \in \mathsf{\Lambda}^+[\Sigma]$	(ab)	$\mathsf{FV}(a) \cup \mathsf{FV}(b)$	$\big(a[y := c]b[y := c]\big)$
$a \in \mathsf{\Lambda}^+[\Sigma \cup \{x\}], x = y$	$(\lambda x.a)$	$\mathsf{FV}(a) \setminus \{x\}$	$(\lambda x.a)$
$a \in \mathsf{\Lambda}^+[\Sigma \cup \{x\}], x \neq y$	$(\lambda x.a)$	$\mathsf{FV}(a) \setminus \{x\}$	$\big(\lambda x.a[y := c]\big)$

where, in the last case, $x \notin \mathsf{FV}(c)$.

With the syntax strictly as above, there is a unique way of *parsing* a term, *i.e.* proving that a given string is indeed a raw λ-term. The substrings of a term which arise in this parsing are called *subterms*. We shall adopt the usual abbreviations:

$$\begin{array}{lll} abc & \text{means} & (ab)c \\ \lambda x.ab & \text{means} & \lambda x.(ab) \\ \lambda xy.a & \text{means} & \lambda x.(\lambda y.a) \end{array}$$

In fact juxtaposition will be used throughout the work to mean application, composition in a category being denoted by a semicolon (;) and written right-handedly.

A raw λ-term a is *closed* if $\mathsf{FV}(a) = \emptyset$. If $\Sigma \subset \Sigma'$ then $\mathsf{\Lambda}^+[\Sigma] \subset \mathsf{\Lambda}^+[\Sigma']$ in a natural way, and the subsets of closed terms are identified. Consequently the set of closed lambda terms is independent of Σ, and is written just $\mathsf{\Lambda}^+$. $\mathsf{\Lambda}^+[\Sigma]$ is then seen as the extension of $\mathsf{\Lambda}^+$ by Σ.

§**1.1.3** The intended meaning of the term $\lambda x.a$ is that function which, to the argument u, assigns the value $a[x := u]$. The process of replacing a (specified) subterm $((\lambda x.a)u)$ by the corresponding $(a[x := u])$ is called a *beta conversion*, and a string of raw λ-terms in which each is obtained from the previous one by a β-conversion is called a *beta transition*.

Since we have defined β-conversion with respect to subterms, β-transition and substitution are compatible, and the equivalence relation generated by the existence of β-conversions is a congruence of the application and abstraction operations. We call it *beta equivalence*.

The quotient sets $\mathsf{\Lambda}[\Sigma]$ and $\mathsf{\Lambda}$ of $\mathsf{\Lambda}^+[\Sigma]$ and $\mathsf{\Lambda}$ under β-equivalence are called the sets of λ-*terms*, or sometimes $\lambda\mathsf{K}\beta$-*terms*.

In $\mathsf{\Lambda}[\Sigma]$ we then have $(\lambda x.a)u = a[x := u]$.

Throughout we shall use Λ (as opposed to $\mathsf{\Lambda}$) for a *model* of the λ-calculus (whatever that is). $\mathsf{\Lambda}$ is called the (*classical*) *closed term algebra* and $\mathsf{\Lambda}[\mathbb{N}]$ the *open term model*.

A raw term from which there is no nontrivial transition is said to be in *normal form*. A raw term from which there is a transition to a normal form, or a term with such a raw representative, *has a normal form*.

§**1.1.4** We shall have occasion to define a variety of special λ-terms, of which the first are

K	$=$	$\lambda xy.x$			constant
S	$=$	$\lambda xyz.xz(yz)$			
I	$=$	$\lambda x.x$	$=$	SKK	identity
Y	$=$	$\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$			fixpoint
$\perp$	$=$	$(\lambda x.xx)(\lambda x.xx)$	$=$	$(\mathsf{SII})(\mathsf{SII}) = \mathsf{YI}$	bottom
P	$=$	$\lambda fgx.g(fx)$	$=$	$\mathsf{S(K(S(S(KS)K)))K}$	composition
Q	$=$	$\lambda fgax.ga(fax)$			fibred composition
$\langle\rangle$	$=$	$\lambda xyz.zxy$			pairing
$\scriptstyle 0$	$=$	$\lambda xy.x$	$=$	K	left projection
$\scriptstyle 1$	$=$	$\lambda xy.y$	$=$	KI	right projection

The combinatory expressions for these things get much longer, so we shall desist from quoting them in future. The reason for switching the arguments of P (in other words not using the better known $\mathsf{B} = \lambda fgx.f(gx)$) will become apparent in §5.1.4. We shall adopt the convention of printing special combinators such as these in sans-serif Roman type, and variables in *italics*.

Warning The letter P (for *P*aul, **P**owerset, *P*air, Product, Pullback) appears in *six* different fonts in this work, apart from those which have already appeared in this sentence, namely: $P\mathsf{P}\mathcal{P}p\mathsf{P}\mathbf{P}$. In §5.1.4 we shall see that $\mathsf{P} = \mathsf{P}$!

Question Why is the S combinator so-called?
Achim Jung (20 February 1991) says that

> "The answer is in Schoenfinkel's paper from 1924 (*Mathematische Annalen* **92**), where combinators are introduced for the first time. S stands for *VerSchmelzungsfunktion* (amalgamation function), because it allows to express that two occurrences of a variable are indeed the same variable, *i.e.* the two occurrences are amalgamated. Schoenfinkel's paper is fun to read; it has been translated into English.
>
> Why did he not use V? There was another combinator called *Vertauschungsfunktion* (exchange function).

§**1.1.5** The consistency of the above formalism is guaranteed by a major classical result known as the *Church-Rosser Theorem*. (Barendregt §11.1)

Proposition β-transitions have the *diamond property* that if $a \to b$ and $a \to c$ then for some d, $b \to d$ and $c \to d$. □

Theorem Λ is not degenerate.

Proof K and S are distinct normal forms and so are not identified under β-equivalence. □

In fact β-reductions may be put in a standard form to give the morphisms of a category with *pushouts* (Barendregt exercise 12.4.4). The "upside-down" diamond property fails.

§**1.1.6** There is a remarkable λ-term, sometimes known as the "paradoxical combinator", which provides a fixed point of (application of) any element.

Lemma For any $f \in \Lambda$, $f(\mathsf{Y}f) = \mathsf{Y}f$.

Proof There is a β-transition from $\mathsf{Y}f$ to $f(\mathsf{Y}f)$. □

There are other such operators for which the transition is in the more natural direction; indeed there is an infinity of fixpoint operators, which are precisely the fixed points of $\mathsf{G} = \lambda yf.f(yf) = \mathsf{SI}$.

§**1.1.7** The importance of the λ-calculus lies in the fact that it can be used to code recursive (computable) functions. Turing [1936] motivated his own definition of computability and subsequently [1937] showed its equivalence with λ-definability.

Briefly (see Barendregt §6.3), *primitive recursive functions* are built up from zero (Z), successor (N), if-zero (?) and product projections together with that if $f(-)$, $g(-,-,-)$ are primitive recursive then so is $h(-,-)$ where $h(0,\vec{n}) = f(\vec{n})$ and $h(k+1,\vec{n}) = g(h(k,\vec{n}),k,\vec{n})$. Then the *recursive functions* are those given by adding *searching* (*minimalisation* in the literature), so if $f(-,-)$ is recursive then so is the *partial* function $g(-)$, where $g(\vec{n})$ is the least k such that $f(i,\vec{n})$ is defined for each $0 \le i \le k$ and $f(k,\vec{n}) = 0$, if there is one; we write $g = \mu f$.

To code all this we of course need some numerals. For instance Church [1941] defined

$$\mathbf{n} = \lambda fx.f^n x$$

so

$$\mathsf{Z} = \mathbf{0} = \mathsf{KI}, \quad \mathsf{N} = \lambda xyz.y(xyz) \quad \text{and} \quad ? = \lambda nxy.n(\mathsf{K}x)y$$

Using these it is an easy matter to code primitive recursive functions. For recursive functions we need a searching operator μ with $g = \mu f$, such as

$$\mu = \lambda f\vec{n}.\mathsf{Y}[\lambda hk.?(fk\vec{n})k(h(\mathsf{N}k))]\mathsf{Z}$$

§**1.1.8** We wish to interpret the λ-calculus in a great variety of circumstances, in which we have far less in the way of logic than is available in the category of sets. In order to do this we reformulate it without mentioning variables (or at least without the variable-binding operator λ). [In fact it *is* possible to do what we intend whilst still carrying variables, by methods suggested in §§3.2.3-5 and 12-13, but this is more complicated. See also [Lambek & Scott, 1986].]

Lemma Λ, the set of closed λ-terms, is generated by $\mathsf{K} = \lambda xy.x$ and $\mathsf{S} = \lambda xyz.xz(yz)$ under application.

Proof We define inductively on the structure of a raw representative the translation

$$\begin{array}{lcll}
[\![\lambda x.x]\!] & = & \mathsf{SKK} & \\
[\![ab]\!] & = & [\![a]\!][\![b]\!] & \\
[\![\lambda x.a]\!] & = & \mathsf{K}[\![a]\!] & \text{if } x \text{ is not free in } a \\
[\![\lambda x.ab]\!] & = & \mathsf{S}[\![\lambda x.a]\!][\![\lambda x.b]\!] & \text{otherwise} \\
[\![\lambda xy.a]\!] & = & [\![\lambda x.[\![\lambda y.a]\!]]\!] &
\end{array}$$

in which the left and right sides are β-equivalent. □

Definition A *combinatory prealgebra* is a set Λ with a binary operation $\bullet$ ("application") and elements K and S satisfying $\mathsf{K}ab = a$ and $\mathsf{S}abc = ac(bc)$ for all $a, b, c \in \Lambda$, where we adopt the usual convention that abc means $(a \bullet b) \bullet c$. It it common to add $\mathsf{I} = \mathsf{SKK}$.

There are then obvious notions of *combinatory transition* and of a *free combinatory prealgebra* on a set of generators ("variables"). This should not, however, be allowed to prejudice our view of what are *morphisms* of models.

Clarke *et al.* [1980] have put this to practical use, building a machine whose instruction set is essentially $\{\mathsf{S}, \mathsf{K}, \mathsf{I}\}$. Curien [1983] considers combinators more closely related to the presentation in terms of cartesian closed categories (§1.3).

§**1.1.9** We shall discuss the assignment of *types* to λ-terms in section 3. There is an obvious way to type I, K and S, namely $P \to P$, $P \to (Q \to P)$ and $[P \to (Q \to R)] \to [(P \to Q) \to (P \to R)]$. In §3.1.6 we shall mention briefly the connection between types and propositions. These types are familiar axioms of propositional calculus. Church's numerals are typed $(P \to P) \to (P \to P)$.

A set Λ with a binary operation $\bullet$ is said to be *combinatory complete* if any definable function $\phi : \Lambda^n \to \Lambda$ (*i.e.* expression involving $\bullet$, elements of Λ and n variables) is expressable as $fx_1x_2...x_n$ for some $f \in \Lambda$, *i.e.* $\forall \vec{a} \in \Lambda^n.f\vec{a} = \phi(\vec{a})$.

Proposition $(\Lambda, \bullet)$ is combinatory complete iff there are $\mathsf{K}, \mathsf{S} \in \Lambda$ such that $(\Lambda, \bullet, \mathsf{K}, \mathsf{S})$ is a combinatory prealgebra.

Proof

[$\Rightarrow$] $x, y \mapsto x$ and $x, y, z \mapsto xz(yz)$ are two such expressions.

[$\Leftarrow$] Use lemma 1.1.8 to express $\lambda\vec{x}.\phi[\vec{x}]$ in terms of K, S. □

§**1.1.10** We can speak of a combinatory prealgebra in any category $\mathcal{C}$ with finite products. It consists of an object $\Lambda \in \mathcal{C}$ and morphisms $\bullet : \Lambda \times \Lambda \to \Lambda$ and $\ulcorner\mathsf{K}\urcorner, \ulcorner\mathsf{S}\urcorner : 1 \rightrightarrows \Lambda$ in $\mathcal{C}$ such that certain diagrams (expressing the various equations) commute, *e.g.*

$$\begin{tikzcd}
(1 \times \Lambda) \times \Lambda \cong \Lambda \times \Lambda \arrow[r, "\pi_0"] \arrow[d, "(\ulcorner\mathsf{K}\urcorner \times 1) \times 1"'] & \Lambda \\
(\Lambda \times \Lambda) \times \Lambda \arrow[r, "\bullet \times 1"] & \Lambda \times \Lambda \arrow[u, "\bullet"']
\end{tikzcd}$$

Moreover if $F : \mathcal{C} \to \mathcal{D}$ is a functor preserving finite products then $(F\Lambda, F\bullet, F\ulcorner\mathsf{K}\urcorner, F\ulcorner\mathsf{S}\urcorner)$ is a combinatory algebra in $\mathcal{D}$ because F (simply by virtue of its being a functor) preserves the validity of the equations. Such a functor is the *global sections functor* $F = \mathcal{C}(1, -) : \mathcal{C} \to \mathbf{Set}$, which we shall sometimes call Γ or $|-|$. This notation with *single* vertical lines will be used for the underlying set (or its cardinality) of an object; *double* verticals will be used for the extension of a type (§1.3.4).

This will enable us to discuss models of the λ-calculus which are not discrete sets.

§**1.1.11** The formulation of §§1.1.2-3 may be extended by adding constants and further equations. In particular, suppose we have a "model" A, whose underlying set we write as $|A|$. Then the term model $\mathsf{\Lambda}[|A|]$, in which we add a constant $\ulcorner a \urcorner$ for each element $a \in A$, has an interpretation in A. This will not however be *faithful*, *i.e.* some distinct terms will be set equal by it. We add to the theory these further equations and write $\mathsf{\Lambda}[A]$ for the resulting term model, which should of course be isomorphic to A.

We may now adjoin a new variable to a model, by making the further addition of a new symbol x. The term model $\mathsf{\Lambda}[A][x]$ we call just $A[x]$. The interpretation of $A \cong \mathsf{\Lambda}[A]$ in $A[x]$ *is* faithful. $A[x]$ has the universal property that any interpretation of A in another model B may be extended uniquely to one of $A[x]$ in which x is taken to an arbitrarily chosen $b \in B$.

To do the same for combinatory prealgebras is a standard easy piece of Universal Algebra.

§**1.1.12** Unfortunately the translation from λ-terms to a combinatory prealgebra does not respect β-equivalence; for example KI and SK are clearly normal forms *qua* combinators, but as λ-terms they are equivalent [to $\lambda xy.y$]. Combinatory prealgebras encapsulate *application*, but fail to represent *abstraction*, for which we require a *canonical* element f whose application to x yields $f[x]$.

By definition in the λ-calculus, $a = b$ implies $\lambda x.a = \lambda x.b$. A λ-term which "begins with a λ", *i.e.* which has a representative of the form $\lambda x.a$, we call *functional*.

Lemma

(a) A λ-term is functional iff it has a representative of the form K, S, $\mathsf{K}a$, $\mathsf{S}a$ or $\mathsf{S}ab$ for some a, b.

(b) If f, g are functional terms (with representatives) in which x is not free, and $fx = gx$ then $f = g$.

(c) A term with a representative beginning with a variable or $\perp$ is not functional.

Proof

(a) The combinatory translation of $\lambda x.a$ involves either the third, fourth or fifth clause in the proof of lemma 1.1.8, so is of one of these forms. Conversely there is a β-transition from each of these forms to a functional λ-term.

(b) W.l.o.g. $f = \lambda x.a$, $g = \lambda x.b$. Then $a = fx = gx = b$.

(c) The forms $\lambda x.a$, $xa_1a_2...a_n$ or $\perp a_1a_2...a_n$ are invariant under transition. Hence by the Church-Rosser theorem (proposition 1.1.5) these three forms are β-inequivalent. □

+§**1.1.13** More generally we can say that a term *has* (at least) $n+1$ *arguments* if whenever it is applied to any (at most) n new variables then the result is functional.

Fact Any λ-term has a representative recursively in the form $\lambda\vec{x}.y\vec{a}$ or $\lambda\vec{x}.(\lambda y.y\vec{a})\vec{b}$, and there is an algorithm to find such a representative for any raw term. □

The first form is called *head normal*, and a term with no such form is *unsolvable*; a raw term is in normal form iff it is recursively head normal. There is no algorithm to determine whether a term has a head normal form or not. The *Böhm tree* of a term is given by developing the head normal form of each subterm, putting $\perp$ instead if it has none; this may of course be infinite. A combinatory algebra is *sensible* [Hyland 1976] if all unsolvable terms are identified.

Lemma A λ-term has exactly n arguments iff it has a representative in the form $\lambda\vec{x}.a$ where a is $y\vec{b}$ (possibly y is amongst the $\vec{x}$) or unsolvable.

Proof By parts (a) and (c) of lemma 1.1.12. □

There are terms with 0, 1, ..., ∞ arguments. Indeed $\lambda x_1...x_n.y$ has exactly n arguments and $\mathsf{YK} = (\lambda x.xx)(\lambda xy.xx)$ has infinitely many.

Corollary The combinators listed in §1.1.4 have exactly as many arguments as λs at the front as they are written there. □

§**1.1.14** We can now complete the reformulation of λ-calculus as a purely equational theory.

Lemma Let $(\Lambda, \bullet, \mathsf{K}, \mathsf{S})$ be a combinatory prealgebra. tfae

(α) For any functional $f, g \in \Lambda$, if $fx = gx$ in $\Lambda[x]$ then $f = g$.

(β) β-equivalent terms are equal.

(γ) The following equations are satisfied.

$$\begin{aligned}
\mathsf{K} &= \mathsf{S(S(KS)(S(KK)K))(K(SKK))}\\
\mathsf{S} &= \mathsf{S(S(KS)(S(K(S(KS)))(S(K(S(KK)))S)))(K(K(SKK)))}\\
\mathsf{S(KK)} &= \mathsf{S(S(KS)(S(KK)(S(KS)K)))(KK)}\\
\mathsf{S(KS)(S(KK))} &= \mathsf{S(KK)(S(S(KS)(S(KK)(SKK)))(K(SKK)))}\\
\mathsf{S(K(S(KS)))(S(KS)(S(KS)))} &= \mathsf{S(S(KS)(S(KK)(S(KS)(S(K(S(KS)))S))))(KS)}
\end{aligned}$$ □

If Λ satisfies these conditions then we say it is a *combinatory algebra*. [In fact most authors use this term in the weaker case, and *lambda algebra* in the stronger.] The importance of part (γ), which is due to Curry, is that we now have a purely equational theory; it now seems unlikely that the form of the equations has any great significance. Part (α) will be used frequently in the constructions of §1.3 and §5.1.

Proposition $(\Lambda[t_1, ..., t_n], \bullet, \mathsf{K}, \mathsf{S})$ is the free combinatory algebra on n generators.

Proof The passage from λ-terms to combinators and back is (β-equivalent to) the identity by lemma 1.1.8. For the converse we use the above lemma. □

§**1.1.15** When we introduced the distinction between prealgebras and algebras in §1.1.12 we would have liked to have said that in the λ-calculus "terms beginning with a λ have the same effect as functions". If, as has been the case so far during this century (though I personally believe this to be a passing heresy), we regard functions as assignments of values to arguments, then this raises questions of *extensionality*.

Proposition There are distinct functional $f, g \in \Lambda$ with $\forall a \in \Lambda. fa = ga$.

Proof Plotkin [1974]. $\square$

Definition A combinatory *algebra* is a *model* if for functional f, g, if $\forall a. fa = ga$ then $f = g$.

Hence $\Lambda[\mathbb{N}]$ is a model and Λ is only an algebra. We shall see that this distinction corresponds to the notion of a *concrete* category.

§**1.1.16** A "stronger" form of extensionality (although it is in fact an independent condition) is the *η-rule* that every element is functional, *i.e.* if $x \notin \mathsf{FV}(a)$ then $\lambda x.ax = a$. We write $\Lambda^\eta[\Sigma]$ for the set of *$\lambda\eta$-terms* in a given set of variables, and Λ^η for the closed $\lambda\eta$-terms. A combinatory algebra or model with this property is called an *η-algebra* or *η-model* respectively. Categorically this corresponds to an *isomorphism* between Λ and Λ^Λ rather than the latter merely being a retract of the former.

We have distinguished between

(i) combinatory *prealgebras* and *algebras* by functionality,

(ii) *algebras* and *models* by concreteness or weak extensionality, and

(iii) *beta* and *eta* by so-called "strong extensionality";

we will distinguish

(iv) *combinatory* and *lambda* by whether we really have function-spaces.

1.2 Retracts, Monoids and Cartesian Closed Categories

§**1.2.1** This section is a miscellany of basic category theory.

Definition Let X be a poset and $D \subset X$ a subset.

(a) D is said to be *directed* if $(\forall d_1, d_2 \in D)(\exists d \in D)(d_1, d_2 \leq d)$.

(b) D is *down-closed* if $(\forall d \in D, x \in X : x \leq d)(x \in D)$

(c) D is an *ideal* if it is down-closed and directed.

Lemma A directed union of directed sets or ideals is directed, respectively an ideal. $\square$

§**1.2.2** Of considerable importance will be *directed unions*, or more generally directed joins (also known as directed sups) in a poset.

Definition (a) We say x is the directed join of $D \subset X$, and write $x = \bigvee^{\uparrow} D$, if

(i) $D \leq x$ (*i.e.* $\forall d \in D. d \leq x$)

(ii) if also $D \leq y$ then $x \leq y$

(iii) D is directed.

Note The use of the arrow implicitly asserts the directedness of the set in question.

Definition (b) We say the function $f : X \to Y$ *preserves directed joins* if $f(\bigvee^{\uparrow} D) = \bigvee^{\uparrow}\{fd : d \in D\}$ for all directed $D \subset X$.

Lemma Let $f : X \to Y$ preserve directed joins. Then f is monotone (preserves order).
Proof Consider the directed set $D = \{x, y\}$ for any $x \le y$. □

Proposition Let X be a poset with a least element $\perp$, and joins for all directed subsets. Let $f : X \to X$ preserve directed joins; then f has a least fixed point.
Proof Put $x_0 = \perp$ and $x_{n+1} = fx_n$. Then $x_0 = \perp \le x_1$ so by induction and monotonicity of f, $x_n = f^n\perp \le f^n x_1 = x_{n+1}$. Hence $D = \{x_n : n \in \mathbb{N}\} \subset X$ is directed. Let $x = \bigvee^{\uparrow} D$. Then $fx = \bigvee^{\uparrow}\{fx_n : n \in \mathbb{N}\}$ by preservation of directed joins, and this is just x. If also $fy = y$ then $x_0 = \perp \le y$, so $x_n = f^n\perp \le f^n y = y$; hence $x \le y$. □

The result that Tarski [1955] proved was that every monotone endofunction of a *complete lattice* has a least (and also a greatest) fixed point.

§**1.2.3** Let $\mathcal{C}$ be a category.

Notation Write the composite of $f : X \to Y$ and $g : Y \to Z$ in $\mathcal{C}$ as $(f\,;g) : X \to Z$.

Warning This is *right-handed* notation (left-handed is slightly more common in Category Theory). The semi-colon is commonly used in programming languages for *sequencing*. Juxtaposition will *always* mean application.

An endomorphism $e : X \to X$ in $\mathcal{C}$ is *idempotent* if e^2 (that is, $e\,;e$) equals e. If $i : U \hookrightarrow X$ and $p : X \twoheadrightarrow U$ in $\mathcal{C}$ have $i\,;p = 1_U$ and $p\,;i = e$ then we say e *splits through* U. This also arises given just i and p with $i\,;p = 1_U$, for then $e = p\,;i$ is automatically idempotent. Some authors introduce several different words for the anatomy of this situation, but we shall indiscriminantly call e or U a *retract* of X. Write $U \lhd X$.

We may regard U as a subobject of X by $i : U \subset X$, since this is (regular, indeed split) mono. In this case the restriction of e to U is just $i : U \hookrightarrow X$ or $1_U : U \to U$; indeed U is precisely the *fixed point set* of e. Categorically, U is the *equaliser* of 1 and e. If we do regard U as a subset we may suppress mention of i and consider $p : X \twoheadrightarrow U$ in place of e; we think of px (for $x \in X$) as x *reduced to* or *reflected in* U.

Alternatively, U is a quotient of X by $p : X \twoheadrightarrow U$, since this is (regular, split) epi. Then U is the *image* of e, which we may describe as the *coequaliser* of 1 and e. i is then a *section* of p.

We shall try to avoid trivial detail when talking about retracts, and the reader will be expected to interpret the context and notation appropriately. Letters such as e (then f), c and r are used for idempotents, p (then q) or π for the surjection and i (then j) or ι for the inclusion.

Notice that when we move to the dual category (in which the arrows are reversed) these concepts either remain the same or are interchanged with one another.

§**1.2.4** Very often there will be an order structure on maps in a category. Retracts which are comparable to the identity in this order play a special role. If $1 \le c = c^2$ we call it a *closure operator*; if $1 \ge c = c^2$ then it is a *coclosure operator*.

Proposition Let $c : X \to X$ be a closure (coclosure) operator on a poset and Y its image. Then the inclusion $i : Y \hookrightarrow X$ is right (left) adjoint to the surjection $p : X \twoheadrightarrow Y$. Conversely if the inclusion of one poset in another has a left (right) adjoint then we have a closure (coclosure). □

Definition Splitting of a coclosure will be of particular importance: we reserve the terms *embedding* and *projection* for this case, with notation $\rightarrowtail$ and $\twoheadrightarrow$.

There are analogous concepts for categories rather than posets: a endofunctor $\mathbf{t}T$ with natural transformations $\eta : 1 \to \mathbf{t}T$ and $\mu : \mathbf{t}T^2 \to \mathbf{t}T$ satisfying certain equations is called a *monad*. It is *idempotent* or a *reflection* if μ is invertible; the image is called a *reflective subcategory*. Dually $\epsilon : \mathbf{t}T \to 1$, $\nu : \mathbf{t}T \to \mathbf{t}T^2$ is a *comonad*, etc

§**1.2.5** In general the composite of two idempotents e, f need not be idempotent.

Lemma

(a) Let e, f be *commuting* idempotents of $X \in \mathcal{C}$. Then $e\,;f$ is idempotent. If further e, f and $e\,;f$ split through U, V, W respectively (which we consider as the fixed point sets), then W is the intersection of U and V.

(b) Let e, f be split idempotents of $X \in \mathcal{C}$. Then the fixed point set of e is contained in that of f iff $e = e\,;f$. □

We write $e \subset f$ if $e = e\,;f = f\,;e$.
What we *shall* mean by *composed retracts* is the situation $X \lhd Y \lhd Z$.

Lemma

(c) If $X \lhd Y \lhd Z$ then $X \lhd Z$, where the inclusion and surjection are the obvious composites.

(d) In this case write e and f for the idempotents on Z corresponding to X and Y respectively; then $e \subset f$.

(e) Conversely $e \subset f$ always arises in this way.

(f) Given $e \subset f$ like this, $e \leq f$ iff X is a coclosure on Y, and $e \geq f$ iff it is a closure. □

§**1.2.6** Preservation by functors is always important.

Proposition Let $e : X \to X$ be an idempotent in $\mathcal{C}$ and $F : \mathcal{C} \to \mathcal{D}$ a covariant or contravariant functor. Then $Fe : FX \to FX$ is also idempotent. Suppose e splits, say through $i : U \hookrightarrow X$, $p : X \twoheadrightarrow U$. Then Fe also splits, through $Fi : FU \hookrightarrow FX$, $Fp : FX \twoheadrightarrow FU$ in the covariant case and $Fp : FU \hookrightarrow FX$, $Fi : FX \twoheadrightarrow FU$ in the contravariant case. □

Also, if the hom-sets are ordered and F preserves the order then it also preserves closures, coclosures, *etc.*

§**1.2.7** Given any category $\mathcal{C}$ we may construct a category $\mathcal{K}(\mathcal{C})$, the *Karoubian completion* of $\mathcal{C}$, in which $\mathcal{C}$ is embedded fully and all idempotents split. $\mathcal{K}(\mathcal{C})$ has *objects* the idempotents $e : X \to X$ in $\mathcal{C}$ and *morphisms* $\alpha : (e : X \to X) \to (f : Y \to Y)$ those $\alpha : X \to Y$ such that $\alpha = e\,;\alpha\,;f$, *i.e.* the diagram

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "\alpha"'] & X \arrow[d, "\alpha"] \\ Y & Y \arrow[l, "f"'] \end{tikzcd}$$

commutes. Composition is as in $\mathcal{C}$ and the identity on $(e : X \to X)$ is e itself. $\mathcal{C}$ is embedded in $\mathcal{K}(\mathcal{C})$ by $X \mapsto (1_X : X \to X)$.

Proposition $\mathcal{C} \to \mathcal{K}(\mathcal{C})$ splits idempotents universally in the following sense. $\mathcal{K}(\mathcal{C})$ has an explicit choice of splittings, so let $\mathcal{D}$ be another category with explicit splittings and $F : \mathcal{C} \to \mathcal{D}$ a functor. Then there is a unique extension to $\mathcal{K}(\mathcal{C}) \to \mathcal{D}$ which preserves splittings. If idempotents already split in $\mathcal{C}$ then the inclusion is an equivalence. □

Since splitting idempotents is equivalent to adjoining finite filtered colimits (or finite cofiltered limits), $\mathcal{K}(\mathcal{C})$ may be found inside the completions of $\mathcal{C}$ under arbitrary or filtered colimits. These are respectively $[\mathcal{C}^{op}, \mathbf{Set}]$ and $\mathsf{Ind}\,\mathcal{C}$ (§2.3.6).

$^+$§**1.2.8** We shall apply the foregoing construction in the particular case of the monoid derived from a model of the λ-calculus. Any monoid M may be regarded as a category with just one object, the morphisms being the elements of M. The unique object of M is taken to the identity in $\mathcal{K}(M)$

Lemma Let e, f be idempotents in a monoid M. Then $e \cong f$ as objects of $\mathcal{K}(M)$ iff there are $u, v \in M$ with $u\,;v = e$, $v\,;u = f$, $u\,;v\,;u = u$ and $v\,;u\,;v = v$ in M. □

Proposition Let M and N be monoids. Then there is a natural equivalence

$$\frac{\text{Functors} \qquad \mathcal{K}(M) \longrightarrow \mathcal{K}(N)}{\text{Semigroup homomorphisms} \qquad M \longrightarrow N}$$

□

Note The point of view which has been adopted in this work is the currently orthodox one that functors must preserve identities as well as composition, which leads to the consideration of the category of retracts. Hoofman [1990] has adopted the opposite attitude, namely that as many of the functors considered in this topic are free extensions to the category of retracts of underlying constructions which preserve composition but not identities we should regard such constructions as respectable mathemaical objects (*semifunctors*). He has shown that much of the development of the well known models of continuous and stable domain theory is simplified by this approach; (essentially) the foregoing result recovers the (identity-preserving) functors.

$^+$§**1.2.9** Monoids, like groups, ought always to be considered to act on something. Under what circumstances can a monoid M be represented as the endomorphisms of a set? If we stretch the meaning of "set", always: M is the endomorphism monoid of $1_M \in \mathcal{K}(M)$.

We can identify the *constants* of M as the elements which are insensitive to precomposition, *i.e.* the $c \in M$ with $\forall m \in M.m\,;c = c$.

Lemma The *regular action* of M on itself, by $m \mapsto (x \mapsto x\,;m)$ restricts to the set C of constants. If the restricted action is faithful, M is a submonoid of the endofunctions of C and we say M *has enough constants*. □

We shall be interested in the monoid of functional elements of a λ-algebra; this will have enough constants iff it is a λ-model.

(One might ask when monoids in the category of vector spaces, *i.e.* rings, have enough constants. It is then appropriate to replace equality by proportionality. We are led to submonoids of the projective matrix monoid, the constants being (projections onto) particular rays, *cf.* PGL, the projective linear group.)

Remark This original form of this result was Cayley's permutation representation of any (abstract) group; another form of it is the Yoneda lemma in category theory.

§**1.2.10** The analogous concept for categories is that there are enough *global elements*, *i.e.* maps from the terminal object to each object, *i.e.* for $f, g : X \rightrightarrows Y$, if $(\forall x : 1 \to X)(x \,;\, f = x \,;\, g)$ then $f = g$. In this case we say 1 is a *generator*. (Many authors gratuitously introduce the contrapositive in this definition.) The dual notion will appear in §2.3.2. Then the global section functor $\mathcal{C}(1, -) : \mathcal{C} \to \mathbf{Set}$ is faithful.

Warning In naïve category theory, a *concrete category* is one whose objects are given sets and whose morphisms are some of the functions between them (*e.g.* groups, fields, topological spaces, but not spaces with homotopies). This may be formalised by saying that there is *a* faithful functor $\mathcal{C} \to \mathbf{Set}$. We shall diverge from this usage by requiring this functor to be $\mathcal{C}(1, -)$; this makes sense in our context because it is always possible to give morphisms with a "constant" value, but the category of groups, for instance, ceases to be concrete.

Proposition Let M be a monoid and $\mathcal{C} \simeq \mathcal{K}(M)$ its Karoubi completion. Then M has enough constants iff $\mathcal{C}$ is concrete. □

§**1.2.11** The analogous concept to directedness in posets is filteredness in categories. By a *diagram* in a category $\mathcal{C}$ we simply mean a functor $d : I \to \mathcal{C}$ from another category I.

Definition A diagram I is said to be *filtered* if

(i) I is nonempty (more properly, inhabited)

(ii) for any two objects $i, j \in I$ there is an object $k \in I$ and morphisms $f : i \to k$ and $g : j \to k$ in I.

(iii) for any parallel pair of morphisms $f, g : i \rightrightarrows j$ in I there is an object $k \in I$ and a morphism $h : j \to k$ with $f \,;\, h = g \,;\, h$.

Examples

(a) Let $I = \omega$, the category with one object for each natural number n and one morphism $n \to m$ iff $n \leq m$. Then any $d : I \to \mathcal{C}$ is filtered.

(b) Let $D \subset X$ be a directed subset of a poset. Consider X as a category with a unique morphism $x \to y$ iff $x \leq y$. Then the inclusion $D \to X$ is a filtered diagram.

(c) Let $e : X \to X$ in $\mathcal{C}$ be idempotent. Then the diagram

$$\begin{tikzcd} X \arrow[r, shift left, "e"] \arrow[r, shift right, "1_X"'] & X \arrow[r, "e"] & X \end{tikzcd}$$

is filtered. □

Lemma Let $d : I \to \mathcal{C}$ be a finite filtered diagram. Then there is a vertex X and an idempotent $e : X \to X$ in d forming (in the manner of example (c)) a subdiagram equivalent to d. Thus d has a colimit iff e splits.

Proof Assign to each even number an object of I, and to each odd number a morphism between previously-named objects, so that each occurs infinitely often. We construct a category and functor $\omega \cup I \to I$ so that $I \to \omega \cup I \to I$ is an equivalence and $\omega \subset \omega \cup I$ full and cofinal; this makes the diagrams $I \to \mathcal{C}$ and $\omega \subset \omega \cup I \to I \to \mathcal{C}$ equivalent in the sense of having the same cones and in particular the same colimit. We incorporate the chosen objects and morphisms into $\omega \cup I$ using axiom (ii):

$$\begin{tikzcd} 0 \arrow[r] & 1 \arrow[r] & \cdots \arrow[r] & 2n-2 \arrow[r] & 2n-1 \arrow[dr] & \\ & & & & & 2n \\ & & & & i_{2n} \arrow[ur] & \end{tikzcd}$$

and (iii) for $i_{2p} \rightrightarrows 2n$:

$$\begin{tikzcd} 0 \arrow[r] & 1 & \cdots 2p-1 \arrow[r] & 2p & \cdots & 2q-1 \arrow[r] & 2q & \cdots 2n-1 \arrow[r] & 2n \arrow[r] & 2n+1 \\ & & i_{2p} \arrow[ur] \arrow[rrr, "f_{2n+1}"] & & & i_{2q} \arrow[ur] & & & & \end{tikzcd}$$

Now let X be any vertex which occurs infinitely often in $\omega \to I$ and f any endomorphism of X which occurs infinitely often as an arrow out of it. f has only finitely many powers; so $f^n = f^{n+m}$ for some $n \geq 0$, $m \geq 1$, and $f^{n+r} = f^{n+r+km}$ for $r \geq 0$, $k \geq 1$. Put $e = f^{n+r}$ where $n + r = km$; then e is the required idempotent. □

Facts

(a) In a category with colimits of filtered diagrams of cardinality less than κ (where κ is an infinite regular cardinal considered as an ordinal) any filtered diagram of *cardinality* κ is equivalent to one of *shape* κ.

(b) In a finite category in which idempotents are split, *every* filtered diagram (however large) has a colimit (*cf.* lemma 2.1.2). □

§**1.2.12** The dual concept to filteredness is of course called *cofilteredness*. The following really amounts to König's lemma.

Lemma Let $X : I \to \mathbf{Set}$ be a cofiltered diagram of nonempty sets.

(a) If each map in the diagram is surjective then the limit is nonempty.

(b) If each set in the diagram is finite then the limit is nonempty.

Proof

[a] By induction on the cardinality of the diagram and using Fact 1.2.11a, w.l.o.g. $I = \kappa^{op}$. Inductively choose $x_\alpha \in X_\alpha$ such that $x_{\alpha+1}$ maps to x_α and $x_\lambda = (x_\alpha : \alpha < \lambda)$ for limit ordinals.

[b] Call $x \in X_i$ *persistent* if for any $j \in I$ there's $k \in I$ and $y \in X_k$ with k beyond i, j in I and y mapping to x, and a similar condition on morphisms. Let $Y_i \subset X_i$ consist of the persistent elements; it is nonempty since otherwise we may choose a point in the diagram beyond points at which each of the finitely many elements of X_i has been exhausted. Then the diagram reduces to one with surjections. □

§**1.2.13** From *infinite colimits* to *finite limits*.

Proposition For a category $\mathcal{C}$ tfae

(α) $\mathcal{C}$ has all finite limits

(β) $\mathcal{C}$ has a terminal object and all pullbacks

(γ) $\mathcal{C}$ has finite products and binary intersections of retracts

(δ) $\mathcal{C}$ has a terminal object, binary products and equalisers.

Proof

$[\alpha \Rightarrow \beta]$ All of these constructions are examples of finite limits.

$[\beta \Rightarrow \gamma]$ Binary products are found by pulling back two terminal projections, and products with more factors by iteration. The image of $e : X \to X$ is the pullback of $(1,1) : X \to X \times X$ against $(1,e) : X \to X \times X$. Intersections are examples of pullbacks.

$[\gamma \Rightarrow \delta]$ It suffices to construct the equaliser of a pair $f, g : A \rightrightarrows B$. First observe that A is a retract of $A \times B$ by $(a,b) \mapsto (a, fb)$. The image of this is usually called the *graph* of the function f and is written A_f; the composite $A_f \to A \times B \to A$ with the product projection is an isomorphism. The equaliser of f and g is now $A_f \cap A_g \subset A \times B \to A$.

$[\delta \Rightarrow \alpha]$ The diagram category $\mathcal{I}$ is essentially expressed in terms of $\mathsf{dom}, \mathsf{cod} : I_1 \rightrightarrows I_0$. The limit of $d : \mathcal{I} \to \mathcal{C}$ is given by the equaliser of the following pair:

$$\begin{array}{ccc} \prod_{i \in I_0} d(i) & \rightrightarrows & \prod_{u \in I_1} d(\mathsf{cod}\, u) \\ \langle x_i : i \in I_0 \rangle & \mapsto & \begin{cases} \langle x_{\mathsf{cod}\, u} : u \in I_1 \rangle \\ \langle u(x_{\mathsf{dom}\, u}) : u \in I_1 \rangle \end{cases} \end{array}$$

□

Such a $\mathcal{C}$ is called a *lex category* (left exact).

§**1.2.14**

Lemma Consider the diagram

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] & \bullet \arrow[r] \arrow[d] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \arrow[r] & \bullet \end{tikzcd}$$

(a) If the two squares are pullback diagrams then so is the rectangle.

(b) If the rectangle and the right hand square are pullbacks then so is the left hand square. □

§**1.2.15** Finite limits and filtered colimits interact nicely.

Proposition

(a) Let $X : I \times J \to \mathbf{Set}$ be a diagram with I filtered and J finite. Then the natural comparison map $\mathsf{colim}_i \lim_j X_{ij} \to \lim_j \mathsf{colim}_i X_{ij}$ is an isomorphism.

(b) The same is true of the category of models of any finitary algebraic theory. □

§**1.2.16** We shall now turn to cartesian closed categories.

Warning In current Computer Science and older Mathematical usage, a category is *cartesian closed* if it as finite products and exponentials. In more recent Category Theory, the term has been redefined to require *all finite limits*. We shall adopt the weaker definition, for reasons which will become apparent in proposition 1.5.11.

Let's do this explicitly.

A *terminal object* $1 \in \mathcal{C}$ has a unique morphism $!_A : A \to 1$ for each $A \in \mathcal{C}$. We usually think of this as the one-point set.

The *product* of $X, Y \in \mathcal{C}$ is an object $X \times Y$ with morphisms $\pi_0 : X \times Y \to X$ and $\pi_1 : X \times Y \to Y$ (*projections* or *components*) which are universal in the sense that if $f : Z \to X$ and $g : Z \to Y$ are any two morphisms of $\mathcal{C}$ then there is a unique morphism $\langle f, g \rangle : Z \to X \times Y$ making the two triangles commute:

$$\begin{tikzcd}
 & X \\
Z \arrow[ur, "f"] \arrow[r, "\langle f,g \rangle"] \arrow[dr, "g"'] & X \times Y \arrow[u, "\pi_0"'] \arrow[d, "\pi_0"] \\
 & Y
\end{tikzcd}$$

We may of course define products with more factors. The terminal object is the product of zero factors, and any finite product may be obtained by iteration.

The *exponential* or *function space* from X to Y is written Y^X or $X \to Y$. The functor $(-)^X$ is defined to be the right adjoint to the product with X, $- \times X$. This means that there is a natural bijection

$$\frac{Z \times X \longrightarrow Y}{Z \longrightarrow Y^X}$$

The operation from top to bottom is *functional abstraction*, λx. If we put $Z = Y^X$ and consider the identity on the bottom, the corresponding thing on the top is the *evaluation map* $\mathsf{ev} : Y^X \times X \to Y$.

§**1.2.17** Finally there is a useful presentation of an object as a retract of its function space.

Notation For an object X of a cartesian closed category write

$$X_0 = X \qquad X_{n+1} = {X_n}^{X_n}$$

Proposition Let X be an object of a cartesian closed category $\mathcal{C}$.

(a) X is inhabited (has a map $1 \to X$) iff $X \lhd X^X$

(b) If the objects of $\mathcal{C}$ carry a partial order, with respect to which X has a least element $\bot$, then $X \lhd X^X$ by a coclosure.

Proof Habitation is necessary because the image of $1_X \in X^X$ in X is an element. Let $\bot \in X$. Then define $X \hookrightarrow X^X$ by $x \mapsto \mathsf{K}x$ and $X^X \twoheadrightarrow X$ by $f \mapsto f\bot$. $\square$

We call this *dropping a variable*.

1.3 Typing Lambda Terms

§**1.3.1** In this section we discuss the typed λ-calculus and its relationship with cartesian closed categories. We also construct the category of retracts, $\mathbf{Retr}(\Lambda)$, of a λ-algebra and show that this enables any term to be typed. Koymans [1984] performs some of these constructions in greater detail. The significance of $\mathbf{Retr}(\Lambda)$ to our work is that any small category of domains is of this form. (We state precisely what can be proved of this assertion in theorem 2.6.12.) Besides the convenience of having a λ-model around, we can also use this formulation to motivate the notion of a continuous type-dependence (definition 5.1.12).

First we construct the monoid of functional elements; throughout Λ is a combinatory algebra.

Definition $\mathbf{1} = \lambda f.\mathsf{PI}(\mathsf{P}f\mathsf{I}) = \lambda fx.fx = \mathsf{S}[\mathsf{K}(\mathsf{SI})]\mathsf{K}$
This is one of Church's numerals (§1.1.7).

Lemma

(a) P is associative in the sense that $\mathsf{P}\alpha(\mathsf{P}\beta\gamma) = \mathsf{P}(\mathsf{P}\alpha\beta)\gamma$ for any $\alpha, \beta, \gamma \in \Lambda$.

(b) The operation $f \mapsto \mathbf{1}f$ is idempotent and its image consists precisely of the functional elements.

Proof

[a] I shall do this kind of thing precisely once, so watch carefully! Either (lemma 1.1.14β)

$$\begin{array}{rcl} \mathsf{P}\alpha(\mathsf{P}\beta\gamma) & = & [(\lambda xyz.y(xz))\alpha][(\lambda xyz.y(xz))\beta\gamma] \\ & = & [\lambda yz.y(\alpha z)][\lambda z.\gamma(\beta z)] \\ & = & \lambda z.[\lambda z.\gamma(\beta z)](\alpha z) \\ & = & \lambda z.\gamma(\beta(\alpha z)) \\ & = & [\lambda xyz.y(xz)]\{\lambda z.\beta(\alpha z)\}\gamma \\ & = & [\lambda xyz.y(xz)]\{[\lambda xyz.y(xz)]\alpha\beta\}\gamma \\ & = & \mathsf{P}(\mathsf{P}\alpha\beta)\gamma \end{array}$$

or (lemma 1.1.14α), since P has three arguments,

$$\begin{array}{rcl} \mathsf{P}\alpha(\mathsf{P}\beta\gamma)z & = & (\mathsf{P}\beta\gamma)(\alpha z) \\ & = & \gamma(\beta(\alpha z)) \\ & = & \gamma(\mathsf{P}\alpha\beta z) \\ & = & \mathsf{P}(\mathsf{P}\alpha\beta)\gamma z \end{array}$$

(b) We now know that the operations $f \mapsto \mathsf{PI}f$ and $f \mapsto \mathsf{P}f\mathsf{I}$ are commuting idempotents, so their composite is idempotent (lemma 1.2.5a). Since P has three arguments, $\mathsf{PI}(\mathsf{P}f\mathsf{I})$ has at least one, so is functional. Conversely any functional element may be seen to be fixed. □

$^{+}$§**1.3.2** Write $\mathcal{M}$ for the set of functional elements of Λ, the image of $f \mapsto \mathsf{PI}(\mathsf{P}f\mathsf{I})$.

Lemma

(a) $\mathcal{M}$ is a retract of Λ

(b) Λ is a retract of $\mathcal{M}$

(c) $\mathcal{M} \subset \Lambda$ is the union of the images of $\ulcorner\mathsf{K}\urcorner : 1 \to \Lambda$, $\ulcorner\mathsf{S}\urcorner : 1 \to \Lambda$, $\mathsf{K}- : \Lambda \to \Lambda$, $\mathsf{S}- : \Lambda \to \Lambda$ and $\mathsf{S}--: \Lambda \times \Lambda \to \Lambda$.

Proof

[a] by lemma 1.3.1, *i.e.* $\mathsf{PI}(\mathsf{P}f\mathsf{I})$

[b] by dropping a variable (1.2.17), *i.e.* $a \mapsto \mathsf{K}a$, $f \mapsto f\bot$

[c] by lemma 1.1.12a □

Proposition

(a) $\mathcal{M}$ carries the structure of a monoid, where I is the identity and P the composition. Specifically, $f\,;g = \mathsf{P}fg$ in right-handed notation.

(b) Λ (as a retract) is the set of constants of $\mathcal{M}$, and the action $\Lambda \times \mathcal{M} \to \Lambda$ is $(a,m) \mapsto ma$.

(c) This action is faithful iff Λ is a combinatory *model*.

Proof

[a] by lemma 1.3.1.

[b] by the above lemma

[c] $\mathcal{M}$ has enough constants iff $(\forall a.m_1 a = m_2 a) \Rightarrow m_1 = m_2$. □

§**1.3.3** In the following paragraphs we combine §1.2.7 and §1.3.2 to construct the *category of retracts*. It is cartesian closed, idempotents split, it has fixpoints and any λ-term is typable in it.

Definition Let Λ be a combinatory algebra and $\mathcal{M}$ its monoid of functional elements. Then $\mathbf{Retr}(\Lambda)$ is the Karoubi completion of $\mathcal{M}$, considered as a category with one object.

Because of the importance of this construction we shall perform it explicitly. An *object* of $\mathbf{Retr}(\Lambda)$ is an idempotent of $\mathcal{M}$, *i.e.* an element $A \in \Lambda$ satisfying $\mathsf{P}AA = A$. Unfortunately *this* operation is not itself idempotent: indeed in general there is no retract of Λ whose image is $\mathsf{ob}\,\mathcal{C} \subset \Lambda$ (*cf.* §1.4.8). The *morphisms* $\alpha : A \to B$ are the elements $\alpha \in \Lambda$ with $\alpha = \mathsf{P}A\alpha = \mathsf{P}\alpha B$. The *identity* on A is A itself, whilst the *composite* of $\alpha : A \to B$ and $\beta : B \to C$ is $\alpha;\beta = \mathsf{P}\alpha\beta : A \to C$.

Notice that we have dropped the category-theoretic convention that the various hom-sets be disjoint, so we do not have functions dom and cod; however it is a straightforward but unenlightening exercise to code these things in if they are required.

For an arbitrary $a \in \Lambda$, we call Aa *a reduced to A*; many constructions are of the form of general or untyped constructions reduced to appropriate types, and we shall frequently quote them as such.

Because of the way $\mathbf{Retr}(\Lambda)$ is given, of course *idempotents split*. So if $\alpha : A \to A$ in $\mathcal{C}$ satisfies $\alpha^2 = \alpha$, *i.e.* $\mathsf{P}\alpha\alpha = \alpha$, then there is an object B and a pair of maps $B \leftrightarrows A$ such that $B \to A \to B$ is the identity and $A \to B \to A$ is α; in fact of course both B and the two maps are represented by α.

§**1.3.4** We now investigate when $\mathbf{Retr}(\Lambda)$ is concrete.

The *terminal object* is $\mathsf{T} = \mathsf{K}\bot$, and this also denotes the *terminal projection*, the unique map $A \to \mathsf{T}$.

Proposition For a combinatory algebra Λ tfae:

(α) Λ is a combinatory model

(β) $\mathcal{M}$ has enough constants

(γ) $\mathbf{Retr}(\Lambda)$ is concrete

Proof By propositions 1.2.10 and 1.3.2c. More explicitly, let $\alpha, \beta : A \rightrightarrows B$ be two maps whose composites with all maps $\mathsf{T} \to A$ are equal. Since α and β are functional and Λ is a *model*, it suffices to show that $\alpha w = \beta w$ for all $w \in \Lambda$. Put $\gamma = \mathsf{K}(Aw)$, then $\gamma : \mathsf{T} \to A$ so by hypothesis $\gamma\,;\alpha = \gamma\,;\beta$. But $\alpha(\gamma\bot) = \alpha(Aw) = \alpha w$ since A is the domain of α. The same holds for β so we are done. □

The type A can be interpreted as the set $\|A\| = \{a : a = Aa\} \subset \Lambda$ and the map $\alpha : A \to B$ as the function $a \mapsto \alpha a$. Note that $\|\mathsf{T}\| = \{\bot\}$ and that every type A has a ("least") element $A\bot$.

§**1.3.5** We shall next show that $\mathbf{Retr}(\Lambda)$ is cartesian closed. For products we have to choose pairing and unpairing combinators $\langle\rangle = \lambda xyz.zxy$, ${}_0 = \lambda xy.x = \mathsf{K}$ and ${}_1 = \lambda xy.y = \mathsf{KI}$.

Lemma $\langle\rangle ab_0 = a$ and $\langle\rangle ab_1 = b$ for all $a, b \in \Lambda$. □

Write $\langle a, b\rangle$, c_0 and c_1 for $(\langle\rangle ab)$, (c_0) and (c_1) respectively, noting carefully the positions of the digits. We do *not* have $\langle c_0, c_1\rangle = c$ (but see §1.4.8 and §2.5.11).

Proposition The *product* $A \times B$ of A and B is $\lambda c.\langle Ac_0, Bc_1\rangle$, which is our abbreviation for $\lambda c.\langle\rangle(A(c_0))(B(c_1))$, with *projections* $\pi_0 = \lambda c.Ac_0$, $\pi_1 = \lambda c.Bc_1$, *i.e.* ${}_0$ and ${}_1$ reduced to (domain $A \times B$ and) codomain A or B. Given $\alpha : D \to A$ and $\beta : D \to B$, $\langle\alpha, \beta\rangle = \lambda d.\langle\alpha d, \beta d\rangle$ is the unique map (*pair*) $D \to A \times B$ making the two triangles commute (1.2.16). □

There is a combinator $\times = \lambda AB.A \times B$ which, when restricted to $\mathsf{ob}\,\mathbf{Retr}(\Lambda) \times \mathsf{ob}\,\mathbf{Retr}(\Lambda) \to \Lambda$, yields (idempotents representing) products. The forgetful functor $|-| : \mathcal{C} \to \mathbf{Set}$ creates finite products and preserves all limits which exist in $\mathcal{C}$.

Corollary Let X be an inhabited object of a cartesian closed category. Then $X \times X$ is a retract of X_3 (see 1.2.17).

Proof We adopt the convention that a, b,... are variables of types involving X once, twice,... in their definition. First $X \lhd X_1$ by dropping a variable. Then

$$\begin{array}{lcl} X \times X & \lhd & X^{\left[(X^X)^X\right]} \\ \langle x, y\rangle & \mapsto & \lambda c.cxy \\ \langle d(\lambda ab.a), d(\lambda ab.b)\rangle & \leftarrow\!\shortmid & d \end{array}$$

So

$$X \times X \quad \lhd \quad X^{\left[(X^X)^X\right]} \quad \lhd \quad \left(X^X\right)^{\left[(X^X)^{(X^X)}\right]} \quad \lhd \quad \left[\left(X^X\right)^{(X^X)}\right]^{\left[(X^X)^{(X^X)}\right]}$$

□

§**1.3.6**

Lemma If $A = \mathsf{P}AA$ and $B = \mathsf{P}BB$ then $\lambda f.\mathsf{P}Af$ and $\lambda f.\mathsf{P}fB$ are commuting idempotents. □

Proposition $\mathbf{Retr}(\Lambda)$ has *function spaces*. $B^A = \lambda f.\mathsf{P}A(\mathsf{P}fB) = \lambda f.\mathsf{P}(\mathsf{P}Af)B$. Given $\alpha : C \times A \to B$ we have the *exponential transpose* $\alpha = \lambda ca.\alpha\langle c, a\rangle : C \to B^A$ and conversely $\alpha = \lambda d.\alpha d_0 d_1$. The *evaluation map* $\mathsf{ev} : B^A \times A \to B$ is given by $\lambda d.C(d_0(Bd_1))$. □

This trick is known as *Currying*, although it was actually discovered by Schönfinkel. Notice that $1 = \mathsf{I}^{\mathsf{I}}$.

Again there are obvious combinators doing these things. $B^A f$ is called *f reduced to domain A and codomain B*, but one should beware that this reduction is *not* functorial (it does not preserve identity and composition).

Theorem $\mathbf{Retr}(\Lambda)$ is cartesian closed. □

§**1.3.7** We would like to think of a "model" of the λ-calculus as an object whose function space is a retract of it in a specified way. Specifically, given a function $f : \Lambda \to \Lambda$, $\lambda(f)$ is its "name" in Λ, and given a name u of a function and an element a, the application ua is in Λ. Hence $\lambda : \Lambda^\Lambda \hookrightarrow \Lambda$ and $\bullet : \Lambda \times \Lambda \to \Lambda$. The β-rule says precisely that λ is the inclusion and $\bullet$ (the exponential transpose of) the surjection of a retract.

Definition Let $\mathcal{E}$ be a cartesian closed category and $\Lambda \in \mathcal{E}$. We say Λ is a *lambda β-algebra* if $\Lambda^\Lambda \lhd \Lambda$ and a *lambda η-algebra* if $\Lambda^\Lambda \cong \Lambda$ in a specified way. If $\mathcal{E}$ is concrete we use the terms *lambda β-model* and *lambda η-model*.

This is the last of the distinctions we summed up in §1.1.15.

Proposition Let Λ be a combinatory algebra. Then I is a lambda β-algebra in $\mathbf{Retr}(\Lambda)$.

Proof I is the type represented by the identity and $\mathcal{M} = \mathsf{I}^\mathsf{I}$ that represented by $\lambda f.\mathsf{PI}(\mathsf{P}f\mathsf{I})$. $\mathcal{M}$ is a retract of I. $\square$

Observe that the endomorphism monoid of $\mathsf{I} \in \mathcal{E}$ is precisely the monoid $\mathcal{M}$ of functional elements, and that this is the set of global elements of I^I.

§**1.3.8** Now we shall discuss what it means for a λ-term to be typable. The basic idea is that for a term fa to (be defined and) have type B, where a has type A, it is necessary that f have type $A \to B$. However besides this function-space construction, there is also a notion of *coercion* if we are to deal satisfactorily with the *untyped* $\lambda\beta$-calculus. Thus the functional element f has type Λ^Λ, but is also to be considered as an element, and to have type Λ.

Coercion first arose in FORTRAN [ANSI 66], where integer or "real" values could be put in a real or complex context. The purpose is to eliminate vexatious but essentially vacuous explicit reinterpretations of values; the analogue in category theory is the "forgetful functor". This is the practical value of "strong typing", used in Mathematics everywhere *except* in the development of Set Theory, that many things can be understood from context.

The compiler is obliged to supply a conversion function; this is reflected categorically by having a canonical map $\mathbb{N} \to \mathbb{R}$ (or $\mathbb{R} \to \mathbb{C}$) which enables morphisms $X \to \mathbb{N}$ and $\mathbb{R} \to Y$ to be composed. This system of canonical maps must be closed under the appropriate categorical notions (in particular composition and exponential), and there must be at most one such map, usually a mono, between any two objects. We consider the subject of coercion in the context of polymorphism in §3.4.9.

The $\subset$ relation in $\mathbf{Retr}(\Lambda)$ (§1.2.5) has appropriate properties.

Lemma $A' \supset A,\ B' \subset B \quad\Rightarrow\quad {B'}^{A'} \subset B^A \quad\Rightarrow\quad B' \subset B$

Proof Easy. The second part follows from $B = \lambda x.B^A(\mathsf{K}x)\bot$. $\square$

$^+$§**1.3.9**

Definition

(a) A *Heyting system of types* is a set V with a binary operation $(\to) : V \times V \to V$ and a partial order $(\subset) \subset V \times V$ such that if $A' \supset A$ and $B' \subset B$ then $(A' \to B') \subset (A \to B)$. Note the contravariance in the first argument! A *homomorphism* $F : V \to W$ of Heyting systems is a function satisfying $FA \to FB = F(A \to B)$ and $A \subset B \Rightarrow FA \subset FB$.

(b) Let $\mathcal{C}$ be a cartesian closed category with *specified* products and exponentials (also projections, evaluation and terminal object). A *coercion structure* on $\mathcal{C}$ is a subcategory $\mathcal{D}$ with the same objects as $\mathcal{C}$ but at most one (*canonical*) map between any two objects, such that (i) if $A \to A'$, $B \to B'$ in $\mathcal{D}$ then $A \times B \to A' \times B'$ in $\mathcal{D}$ and (ii) if $A' \to A$, $B \to B'$ in $\mathcal{D}$ then $B^A \to {B'}^{A'}$ in $\mathcal{D}$.

Examples

(a) By default the *free* Heyting system on countably many generators is commonly meant.

(b) Let $\mathcal{C}$ be a cartesian closed category with (specified products and exponentials and) a coercion structure $\mathcal{D}$. Then $(V, \to, \subset)$ is the *underlying Heyting system* on $\mathcal{C}$, where $V = \mathsf{ob}\,\mathcal{C}$, $A \to B = B^A$ and $A \subset B$ iff there is a canonical map $A \to B$.

(c) Let $\mathcal{C} \to \mathcal{C}'$ be a functor preserving specified products and exponentials and canonical maps. Then there is a homomorphism of their corresponding underlying Heyting systems.

(d) In example (b) suppose $\mathcal{D}$ consists only of identities. Then $A \subset B$ iff $A = B$, and $(V, \to, =)$ is the *underlying discrete Heyting system* on $\mathcal{C}$, written $\|\mathcal{C}\|$.

(e) Let Λ be a combinatory algebra. Then $V = \mathsf{ob}\,\mathbf{Retr}(\Lambda)$, $A \to B = B^A$, $A \subset B \iff A = \mathsf{P}AB = \mathsf{P}BA$ is the *retract* coercion structure on $\mathbf{Retr}(\Lambda)$, the canonical map being $A : A \to B$.

$^+$§**1.3.10** Given a Heyting system $(V, \to, \subset)$ we may say when a λ-term $a \in \Lambda[t_1, ..., t_n]$ is typable w.r.t. V when $t_1, ..., t_n$ are assigned types $A_1, ..., A_n$ respectively:

(i) t_i has type A_i

(ii) If f has type $A \to B$, and a has type A, then fa has type B

(iii) If $b \in \Lambda[t_1, ..., t_n, x]$ has type B when x is assigned type A, then $\lambda x.b$ has type $A \to B$

(iv) If a has type A' and $A' \subset A$ then a also has type A.

Now we shall define a cartesian closed category $\mathbf{Type}(V)$ with a coercion structure.

The *objects* are finite strings of elements of V, written as formal products (the empty string is written 1); we shall write $\vec{A}$ for $A_1 \times ... \times A_n$. The *morphisms* $\vec{A} \to \vec{B}$, where $\vec{A}$ and $\vec{B}$ have lengths n, m respectively, are the strings of m $\Lambda^+[\vec{t}]$ terms of types $\vec{B}$ where the variables have types $\vec{A}$, identified under the rules

$$\frac{\begin{array}{c}[x:A]\\ \vdots \\ b : B \quad a : A\end{array}}{(\lambda x.b)a = b[x := a]}\,(\beta) \qquad \frac{f : A \to B \quad x \notin \mathsf{FV}(f)}{f = \lambda x.fx}\,(\eta)$$

The *identity* on $\vec{A}$ is $\vec{x}$ and *composition* is given by substitution. The *coercion structure* is also given by variables.

The *terminal object* is the empty string, and the *product* of two strings their concatenation; the *projections* are just appropriate substrings of variables.

The *exponential* is

$$A_1 \times ... \times A_n \to B_1 \times ... \times B_m \quad = \quad (A_1 \to (A_2 \to ...(A_n \to B_1)...)) \times ... \times (A_1 \to ...B_m)$$

The *exponential transpose* $\mathsf{hom}(\vec{A} \times X, \vec{B}) \to \mathsf{hom}(\vec{A}, \vec{B}^X)$ is λx. We need the *typed* η-rule as well as the β-rule to make this a bijection; notice that it applies *only* to *functional* types. *Evaluation* is given by application.

Given a homomorphism $F : V \to W$ of Heyting systems, $\mathbf{Type}(F)$ on objects applies F to each factor in the string, and on morphisms reinterprets terms.

Proposition $\mathbf{Type}(V)$ is a cartesian closed category and $\mathbf{Type}(F) : \mathbf{Type}(V) \to \mathbf{Type}(W)$ a functor which preserves products, exponentials and canonical maps.

Proof The functor is well-defined on morphisms because any term which is V-typable by A is W-typable by FA, and any equality which holds for V also holds for W. It preserves products, exponentials and canonical maps by construction. $\square$

$^{+}$§**1.3.11** We now compare the cartesian closed categories derived from the naturally occurring Heyting systems on given cartesian closed categories.

Proposition The functor **Type** gives the *free cartesian closed category with coercion* on a Heyting system of types, *i.e.* it is left adjoint to the underlying Heyting system functor, $\|-\|$.

Proof

- $[\eta]$ Let $(V,\to,\subset)$ be a Heyting system. $\|\mathbf{Type}(V)\|$ has elements strings of V-elements, the unit being the inclusion of singleton strings; $\to$ is given in the manner of the exponential in $\mathbf{Type}(V)$ and two strings are coercible iff they have the same length and are elementwise coercible.

- $[\epsilon]$ Let $\mathcal{C}$ be a cartesian closed category with coercion. $\mathbf{Type}(\|\mathcal{C}\|)$ has new products and only the λ-*definable* morphisms (ϵ being the inclusion of these). □

The adjunction is idempotent in the pseudo sense that the inclusion of $\mathbf{Type}(V)$ in $\mathbf{Type}(\|\mathbf{Type}(V)\|)$ is an equivalence.

$^{+}$§**1.3.12** Now we apply this to a λ-model.

Lemma Let V be a Heyting system with an element Λ such that $\Lambda\to\Lambda\subset\Lambda$. Then every raw λ-term has type Λ.

Proof In the notation of §1.2.17, $\Lambda_n\subset\Lambda_m$ for any $n\geq m$. Define the *degree* of a raw λ-term to be the depth of occurrences of application, so that $\partial x=0$, $\partial(ab)=\mathsf{max}(\partial a+1,\partial b)$, $\partial(\lambda x.a)=\partial a$. If a raw λ-term has type Λ_k when its free variables are assigned types Λ_m then it has type Λ_{k+r} when we replace m by n with $n\geq m+r$. Claim that any raw term a has type Λ in this case if $m\geq\partial a$. This is so for variables. For (ab), a may be retyped as Λ^Λ and b as Λ. Finally $\lambda x.a$ has type $\Lambda^\Lambda\subset\Lambda$. □

Proposition Let $\mathcal{C}$ be a cartesian closed category with an object Λ for which Λ^Λ is a retract of Λ in a specified way. Then Λ carries a combinatory algebra structure in $\mathcal{C}$, which is preserved by the global section functor $|-|=\mathcal{C}(1,-)$.

Proof First replace $\mathcal{C}$ by an equivalent category $\mathcal{C}'$ with *specified* products and exponentials (the objects are terms in $\times$, $\to$ and the objects of $\mathcal{C}$, the hom-sets are those for the isomorphic objects of $\mathcal{C}$). Now let $\mathcal{D}$ be the coercion structure on $\mathcal{C}'$ generated by the inclusion $\Lambda^\Lambda\hookrightarrow\Lambda$ (these are of course *unequal* objects of $\mathcal{C}'$); since the objects of $\mathcal{C}'$ are uniquely expressable as products and exponentials of $\mathcal{C}$ objects, there is indeed at most one (deduction of a) canonical map between any two objects.

Let V be the underlying Heyting system. This satisfies the hypotheses of the lemma, so any λ-term has type Λ. In particular we may interpret $\bullet$, K and S and verify that these give a combinatory algebra structure on Λ. Such structure is preserved by any functor which preserves products (§1.1.10). □

§**1.3.13** Finally we show that the constructions of §1.3.7 and 12 are inverse.

Theorem

(a) Let Λ be a combinatory model. Then Λ is isomorphic to the global sections of the combinatory model structure on the lambda model $\Lambda=\mathsf{I}\in\mathbf{Retr}(\Lambda)$.

(b) Let Λ be a lambda model in a cartesian closed category $\mathcal{C}$, such that every object of $\mathcal{C}$ is a retract of Λ and idempotents split in $\mathcal{C}$. Then $\mathbf{Retr}(|\Lambda|)$ is equivalent to $\mathcal{C}$.

Proof

[a] Following through the constructions, each $a \in \Lambda$ is interpreted by something of the same name.

[b] Every object of $\mathcal{C}$ is represented by a retract of Λ, which is an idempotent global element of Λ^Λ and hence a global element of Λ. $\square$

Question What information does $\mathbf{Retr}(\Lambda)$ give about equality of terms in Λ?

We return in §5.6.10 to the question of repeating the **Retr** construction for the *internal* combinatory algebra in $\mathbf{Retr}(\Lambda)$.

1.4 The Graph Model

§**1.4.1** We have now settled on the definition of a model of the λ-calculus as an object of a cartesian closed category whose function space is a retract of it in a canonical way. Of course there are trivial cardinality reasons why this cannot happen in the category of sets, so this view of the λ-calculus remained dormant for a long time and at one stage it was even doubted whether there was any model of mathematical interest. Then in 1969 Scott showed that the function-spaces of what subsequently became known as continuous lattices were essentially no larger than the original objects, opening the way for a continuous lattice model of the λ-calculus known as D_∞. The techniques which were developed from this form the subject of Chapter II.

Then Plotkin (and later independently Scott), *circa* 1972, showed how to make $P\omega$ (the powerset of the natural numbers) into a model of the λ-calculus, though Scott [1979] traces earlier occurrences of the ideas and sketches a hypothetical Master's thesis which he claims could have been written in 1928. The account given here is a highly potted version of Scott [1976]. The category of retracts of this model is precisely the category of (countably-based) continuous lattices and Scott-continuous maps.

The definition $D \cong D \to D$ of an η-model is an example of a *recursive domain equation*. D_∞ was the first (external, categorical) solution of such a problem, and this technique was developed by Smyth and Plotkin [1978]. This is rather peculiar in that the exponential functor is contravariant in the first argument, and so an auxiliary class of morphisms must be introduced to render the function space construction functorial.

$P\omega$ also enables domain equations to be solved, but this time by internalising the constructions (again in an apparently non-functorial way); the required solution is then obtained simply by use of the fixpoint combinator. One of our aims is the explanation of why these two techniques work.

In this section we describe the $P\omega$ model, which has the property that the Y combinator really does give *least* fixed points (with respect to its natural order).

§**1.4.2** In order to obtain this property for Y, it is necessary to choose particular explicit bijections among $\mathbb{N}$ itself, its set $\mathbb{N} \times \mathbb{N}$ of ordered pairs and its set $P_f\mathbb{N}$ of finite subsets.

These are "cross-diagonal" and "binary" counting:

$$\begin{array}{rcl} \langle n, m\rangle & = & \frac{1}{2}(n+m)(n+m+1)+m \\ a & = & \sum\{2^n : n \in a\} \\ \mathrm{e}_m & = & \{n : \text{the } nth \text{ binary digit of } m \text{ is } 1\} \end{array}$$

$$\begin{array}{ccccccc} & 20 & & & & & \\ & 14 & 19 & & & & \\ & 9 & 13 & 18 & & \cdots & \\ m & 5 & 8 & 12 & 17 & & \\ \uparrow & 2 & 4 & 7 & 11 & 16 & \\ & 0 & 1 & 3 & 6 & 10 & 15 \\ & & \to & n & & & \end{array}
\qquad
\begin{array}{rl} \emptyset & = 0 \\ \{0\} & = 1 \\ \{1\} & = 2 \\ \{1, 0\} & = 3 \\ \{2\} & = 4 \\ \{2, \quad 0\} & = 5 \\ \{2, 1 \quad\} & = 6 \\ \{2, 1, 0\} & = 7 \\ \cdots & \end{array}$$

Lemma

(a) $m \leq \langle n, m \rangle$

(b) $n \in \mathrm{e}_m$ implies $n < m$. □

§**1.4.3** $P\omega$ carries an order (setwise inclusion), and a topology, the *Scott topology*, which we shall discuss in detail in §2.1.4. The sets $\uparrow \mathrm{e}_m = \{a \in P\omega : \mathrm{e}_m \subset a\} \subset P\omega$ form a base for this topology, *i.e.* any open set is a union of sets of this form.

Lemma A function $f : P\omega \to P\omega$ is continuous iff for all $a \in P\omega$ we have

$$f(a) = \bigcup^{\uparrow}\{f(\mathrm{e}_m) : \mathrm{e}_m \subset a\}$$

Proof

[⇒] Let $n \in f(a)$, so $a \in f^{-1}(V)$ where $V = \mathrm{e}_{2^n}$. Hence by continuity there is a basic $a \in \uparrow \mathrm{e}_m \subset f^{-1}(V)$. Then $n \in f(\mathrm{e}_m)$ and $\mathrm{e}_m \subset a$, n lies in the right-hand side and we have $\subset$. $\supset$ is trivial.

[⇐] The sets $V = \mathrm{e}_{2^n}$ are subbasic (generate the topology by finite intersection and arbitrary union), and it suffices to show that $f^{-1}(V)$ is open. Let $a \in f^{-1}(V)$, *i.e.* $n \in f(a)$; then $n \in f(\mathrm{e}_m)$ for some $\mathrm{e}_m \subset a$, *i.e.* $a \in \uparrow \mathrm{e}_m \subset f^{-1}(V)$. □

§**1.4.4** It suffices to define a continuous function f on these finite sets, and any *monotone* such definition will do.

Consequently a continuous function is determined by the set

$$\{\langle n, m \rangle : n \in f(\mathrm{e}_m)\}$$

which, by the identification of pairing above, is an element of $P\omega$.

Lemma

(a) With this coding, application is given as

$$fa = \{n : (\exists m)(\mathrm{e}_m \subset a \wedge \langle n, m \rangle \in f)\}$$

(b) The composite of these two operations,

$$u \mapsto \{\langle n, m \rangle : (\exists m')(\mathrm{e}_{m'} \subset \mathrm{e}_m \wedge \langle n, m' \rangle \in u)\}$$

is a closure operator whose image consists of the codes of continuous functions. □

Proposition $P\omega$ is a β-model in **Sp**. □

Strictly speaking **Sp** is not cartesian closed but $\mathbf{Retr}(P\omega)$ is a full subcategory of **Sp** known as $\mathbf{ContLat}_\omega$; in theorem 2.3.8 we shall see that it consists precisely of the countably based injective spaces. The inclusion and surjection maps are called "graph" and "fun", hence the alternative name *graph model* for $P\omega$.

§**1.4.5** Our interest in the $P\omega$ model lies partly in the fact that its retracts are the continuous lattices and that it is a retract of any other continuous β-model but also in that the Y combinator gives exactly *least* fixed points.

Proposition Let $\mathsf{Y} = [\![\lambda f(\lambda x.f(xx))(\lambda x.f(xx))]\!]$. Then $\mathsf{Y}f$ is the least fixpoint of $f : P\omega \to P\omega$.

Proof (Scott) Put $w = \lambda x.f(xx)$ and let $fa = a$. To show $ww \subset a$ it suffices by continuity to show that $\mathrm{e}_m \subset w$ implies $\mathrm{e}_m\mathrm{e}_m \subset a$. This we show by induction on m.

Suppose then that the result holds for $m' < m$, and that $\mathrm{e}_m \subset w$ with $k \in \mathrm{e}_m\mathrm{e}_m$: we want $k \in a$. But by the definition of application, there is some m' with $\langle k, m'\rangle \in \mathrm{e}_m$ and $\mathrm{e}_{m'} \subset \mathrm{e}_m$. But $m' \leq \langle k, m'\rangle < m$ by lemma 1.4.2, and $\mathrm{e}_{m'} \subset w$, so by hypothesis $\mathrm{e}_{m'}\mathrm{e}_{m'} \subset a$. Note also that $\langle k, m'\rangle \in w$ and w is defined by λ-abstraction, so $k \in w\mathrm{e}_{m'} = f(\mathrm{e}_{m'}\mathrm{e}_{m'})$. By monotonicity $f(\mathrm{e}_{m'}\mathrm{e}_{m'}) \subset fa = a$ so $k \in a$ as required. □

The use of the properties of the identifications is essential, as shown by Baeten and Boerboom [1979]. The crucial point, however, was identified much earlier by Park [1969]; it is that the order on $\mathbb{N}$ make tuples more complicated than their components.

§**1.4.6** On the combinatory algebra structure of $P\omega$ we may as before construct the cartesian closed category $\mathbf{Retr}(P\omega)$ of types. In the previous discussion of this it was remarked that the various constructions (product, exponential, *etc.*) were performed by means of combinators. For example

$$\begin{aligned} \times &= \lambda ABz.\langle Az_0, Bz_1\rangle \\ \to &= \lambda ABf.\mathsf{P}A(\mathsf{P}fB) \end{aligned}$$

when restricted to types (idempotents) yields their product or exponential.

A kind of "sum" may also be constructed using the $P\omega$-combinator

$$\mathsf{cond} \quad = \quad \lambda txy.\begin{cases} \emptyset \text{ if } t = \emptyset \\ x \text{ if } t = \{0\} \\ y \text{ if } 0 \notin t \neq \emptyset \\ \omega \text{otherwise} \end{cases}$$

so

$$A + B \quad = \quad \lambda z.\mathsf{cond}\ z_0\ \langle\{0\}, Az_1\rangle\ \langle\{1\}, Bz_1\rangle$$

The effect of this on lattices is to take their disjoint union and add new top and bottom elements (*cf.* lemma 2.6.7). It is most certainly not a binary coproduct, but we shall see in §5.7.4 what it really means.

Recall that retracts are ordered by inclusion of their underlying sets: $A \subset B$ iff $\|A\| = \{a : a = Aa\} \subset \|B\|$; equivalently $A = \mathsf{P}AB = \mathsf{P}BA$. This is *not* the same as the order inherited as elements of $P\omega$. We also say a retract is *strict* if $A\bot = \bot$; of course since Y gives *least* fixed points this really is the bottom element $\emptyset$.

§**1.4.7** Using these operators we may write down domain equations, for instance $N = 1 + N$ for the "natural numbers" and $D = A + (D \to D)$ for a model of the λ-calculus with atoms.

Proposition Let $F \in P\omega$ preserve retracts (so $\mathsf{P}AA = A$ implies $\mathsf{P}(FA)(FA) = FA$). Then $\mathsf{Y}F$ is a retract. If also F preserves strictness and $\subset$ then $\|\mathsf{Y}F\|$ is the bilimit of the $\|F^n\bot\|$.

Proof By proposition 1.4.5 $\mathsf{Y}F = \bigvee^{\uparrow}(F^n\bot)$. By preservation of retracts each term is a retract and by continuity of composition so is the limit. This is *not* a combinatory fact about Y: it depends on proposition 1.4.5. We shall introduce bilimits in proposition 2.2.4 and explain this more fully in §5.7. □

Our two equations may now be "solved" as $N = \mathsf{Y}(\lambda n.1 + n)$ and $D = \mathsf{Y}(\lambda d.A + d \to d)$. This may be used to provide domains for more complicated programming languages than the

λ-calculus, in which objects may be numbers, functions, trees, ... whose values may themselves be arbitrary objects. We simply write down (essentially) the Backus-Naur form of the definition of an expression and apply the Y combinator to it; the syntactic operations in the language will also correspond to semantic operations on the recursively defined type.

This is all very well (and it will be clear that this internal Scott method gives the same answers as the external Plotkin method) but we can hardly claim to have explained what it all means, and surely if we are to use the word "semantics" for these techniques it is incumbent upon us to do so.

§**1.4.8** What can we say of the collection of retracts? It's easy to see that it forms a complete lattice, but Ershov has shown that it is not continuous and so is not the image of a retract [Hosono & Sato 1977]. Scott's answer to this was to restrict attention to the *closure operators*, *i.e.* the continuous *inflationary* idempotents ($1 \leq c = c^2$).

Proposition The category of closure operators and functions commuting with them is cartesian closed, and there is a type whose *elements* correspond bijectively to the *types*.

Proof We have to adopt a different pairing combinator, specifically

$$\begin{aligned} [x, y] &= \{2n : n \in x\} \cup \{2n+1 : n \in y\} \\ u_0 &= \{n : 2n \in u\} \\ u_1 &= \{n : 2n+1 \in u\} \end{aligned}$$

which satisfies not only $[x, y]_0 = x$ and $[x, y]_1 = y$ but also $u = [u_0, u_1]$, *i.e.* it is *surjective*. With this we construct products and exponentials as before.

The type of types is constructed in the obvious way: $\mathsf{V} = \lambda A. \bigcup A^n$ which is (and has values which are) continuous and idempotent by continuity of composition. it is inflationary because A itself occurs in the union and its values are because the identity (A^0) occurs. □

This category actually consists of the countably based *algebraic* lattices (proposition 2.3.4).

This "type of types" combinator V enables us to interpret polymorphism (where the types are variable), which will be discussed in chapter III. However once again we have begged the question as to what it all means. It is my opinion the use of *closure* operators here is also a mistake.

§**1.4.9** The intuitive reason why Y should give the least fixed point is that $\mathsf{Y}f = f(\mathsf{Y}f) = f^2(\mathsf{Y}f) = ...$, and so is the limit of the sequence $\bot$, $f\bot$, $f^2\bot$, ... More generally, we may approximate terms syntactically by substituting $\bot$ for subterms of β-equivalent terms; such approximants are called *Böhm trees* if they are in normal form (§1.1.13), and a model is *continuous* if (it is sensible, *i.e.* identifies unsolvable terms to $\bot$, and) the interpretation of a term is the directed sup of those of its Böhm approximants. Thus in particular Y is the least fixed point combinator in a continuous model.

Proposition $P\omega$ is a continuous model. □

1.5 Fixpoint Properties in Categories

+§**1.5.1** Having now agreed on the definition of a model of the λ-calculus, it is appropriate the discuss the various forms which one might adopt for an axiom demanding fixpoints for all objects in a category, and to investigate the elementary consequences of these. Let $\mathcal{E}$ be throughout a cartesian closed category.

Recall that a lambda β- (respectively η-) algebra is an object $\Lambda \in \mathcal{E}$ with $\Lambda^\Lambda \lhd \Lambda$ (respectively $\Lambda^\Lambda \cong \Lambda$).

Define the following *reflexivity conditions* on $A \in \mathsf{ob}\,\mathcal{E}$:

[R_5] A is *η-reflexive* if $A \lhd \Lambda$ for some η-algebra Λ in $\mathcal{E}$.

[R_4] A is *β-reflexive* if $A \lhd \Lambda$ for some β-algebra Λ in $\mathcal{E}$.

[R_3] A is *reflexive* if $A^U \lhd U$ for some object U of $\mathcal{E}$.

[R_2] A *has internal fixpoints* if there's a map $\mathsf{Y}_A : A^A \to A$ such that the two composites below are equal:

$$A^A \xrightarrow{\langle \ulcorner \mathsf{id} \urcorner, \mathsf{Y}_A \rangle} A^A \times A \underset{\pi_1}{\overset{\mathsf{ev}_A}{\rightrightarrows}} A$$

[R_1] A *has external fixpoints* if for every $f : A \to A$ there's some $a : 1 \to A$ such that $a\,; f = a$.

[R_0] A is *inhabited* if there's some map $1 \to A$ from the terminal object of $\mathcal{E}$.

A category in which every object satisfies R_4 or R_5 is said to have *enough models of the lambda calculus*. If one model Λ suffices for the whole category then it is *saturated* (see §2.6.1) and we have a subcategory of $\mathbf{Retr}(\Lambda)$; clearly this will not be the case for *large* categories (in the sense of Definition 2.5.2).

$^+$§1.5.2

Proposition Let $F : \mathcal{E} \to \mathcal{F}$ be a functor preserving products and exponentials; then β- and η-algebras and each of R_0 to R_5 are preserved. If in addition F is full and faithful then η-, β-algebras, R_2, R_1 and R_0 are reflected.

Proof These follow easily from the form of the definitions. $\square$

$^+$§1.5.3

Lemma Any finite power of an η- or β-algebra or of an $\eta-$ or β-reflexive object has the same property.

Proof Let $\Lambda^\Lambda \lhd \Lambda$. By repeated Currying we have $\Lambda^{\Lambda^n} \lhd \Lambda$; for $\Lambda^{\Lambda^{n+1}} \cong (\Lambda^\Lambda)^{\Lambda^n} \lhd \Lambda^{\Lambda^n}$, and so we have this by induction. So $(\Lambda^n)^{(\Lambda^n)} \cong (\Lambda^{\Lambda^n})^n \lhd \Lambda^n$. Similarly with isomorphisms. Also if $A \lhd \Lambda$ then $A^n \lhd \Lambda^n$. $\square$

$^+$§1.5.4 We now show the implications between these properties.

Lemma Each of R_5, R_4, R_2 and R_1 implies R_0, which is equivalent to $A \lhd A^A$.

Proof These may be seen by consideration of the identity as an element of Λ^Λ or A^A; the remaining case ($R_3 \Rightarrow R_0$) will follow shortly. If A has an element (which we may call $\perp$) then A is a retract of A^A by dropping a variable (§1.2.17), *i.e.* $a \mapsto \lambda b.a$ and $f \mapsto f\perp$. $\square$

$^+$§1.5.5

Lemma Each property is stable under retracts. R_5 and R_4 are equivalent to reflexivity by an η- or β-algebra respectively.

Proof These follow because retracts are preserved by any covariant or contravariant functor, in particular composition and either side of exponentiation. Let $B \lhd A$.

[R_5, R_4] $B \lhd A \lhd \Lambda$ implies $B \lhd \Lambda$.

[R_3] $B^U \lhd A^U \lhd U$.

[R_2] This requires some thought, but the composite $B^B \hookrightarrow A^A \to A \twoheadrightarrow B$ will be found to work.

[R_1] The external version of the same thing. Let $f : B \to B$, and put $g : A \twoheadrightarrow B \to B \hookrightarrow A$ for the composite with the surjection and inclusion. We recover f as $B \hookrightarrow A \to A \twoheadrightarrow B$. Let $a \in A$ with $ga = a$; then the image of a under the surjection is fixed by f.

[R_0] by composition with the surjection.

[$R_5 \Rightarrow R_3^\eta$, $R_4 \Rightarrow R_3^\beta$] $A^\Lambda \lhd \Lambda^\Lambda \lhd \Lambda$

[$R_3^\beta \Rightarrow R_4$, $R_3^\eta \Rightarrow R_5$] $A \lhd A^U$ by dropping a variable, *i.e.* $a \mapsto \mathsf{K}a$ and $f \mapsto f\bot$ where $\bot \in U$. Then $A \lhd U$. □

$^+$§**1.5.6** The numbers are justified by the

Proposition Each property implies the next.

Proof It remains only to show that $R_3 \Rightarrow R_2$. Let $c : A^A \times A^U \to A^U$ be the obvious composition map and $\Delta : U \to U \times U$ the diagonal. Write $u : 1 \to A^U$ for the exponential transpose of

$$U \xrightarrow{\Delta} U \times U \xrightarrow{\overline{p}} A$$

where $\overline{p}$ is the exponential transpose of the surjection $p : U \twoheadrightarrow A^U$, the inclusion $A^U \hookrightarrow U$ being called i. We might say that u is $\lambda x.[p(x)](x)$. Then let $\mathsf{Y}_A : A^A \to A$ be

$$A^A \times 1 \xrightarrow{1 \times u} A^A \times A^U \xrightarrow{c} A^U \xhookrightarrow{i} U \xrightarrow{\Delta} U \times U \xrightarrow{\overline{p}} A$$

This has the required property. □

Of course this is just a specialisation of the Y combinator, and the "bottom" element constructed for Λ by this process is just $(\lambda x.xx)(\lambda x.xx)$. We have in particular shown that $\mathbf{Retr}(\Lambda)$ has internal fixpoints; it will follow shortly that this means that it cannot have binary coproducts, equalisers or intersections of subobjects.

$^+$§**1.5.7** We can force these properties by putting strong completeness conditions on $\mathcal{E}$; this is based on an idea of Martin Hyland:

Proposition Suppose $\mathcal{E}$ has products over indexing sets of cardinality up to that of the *object* set of $\mathcal{E}$. Then it has a β-algebra of which any inhabited object is a retract.

Proof Let $I \subset \mathsf{ob}\,\mathcal{E}$ be the set of inhabited objects and $\Lambda = \prod_{A \in I} A$ their product. Clearly $\Lambda^\Lambda \in I$ because $1_\Lambda \in \Lambda^\Lambda$. Choose a point a for each inhabited object $A \in I$. It suffices that any $B \in I$ be a retract of Λ, for then $\Lambda^\Lambda \lhd \Lambda$ also. The surjection $\Lambda \twoheadrightarrow B$ is simply the product projection, since B is one of the factors. The image of $b \in B$ under the inclusion is $(u_A(b) : A \in I)$, where

$$u_A(b) \quad = \quad \begin{cases} b \text{if } A = B \\ a (\text{the chosen point of } A) \text{ otherwise} \end{cases}$$

□

The peculiar thing about this result is not really its use of the Axiom of Choice: this "axiom" is commonly found to be a theorem when results using it are *applied*, because there is usually enough structure in the presentation. The difficulty lies in the dichotomy in the definition of the retract, and in the notion of equality of objects.

Example Let Λ be a domain with $\Lambda \cong \Lambda \times \Lambda \cong \Lambda^\Lambda$. We shall construct such a device in §2.5.11. Then $\mathcal{E} = \{1, \Lambda\}$, with the obvious morphisms, satisfies the conditions of the Proposition: since there are only two objects we need only binary products.

§**1.5.8** We cannot ask for bigger products than this because of the well-known result due to Peter Freyd:

Proposition Let $\mathcal{C}$ be a category with at most κ morphisms. If $\mathcal{C}$ has κ-indexed products then it is a complete lattice.

Proof Suppose $f, g : X \rightrightarrows Y$ are distinct. Then $|\mathcal{C}(X, Y^\kappa)| = |\mathcal{C}(X,Y)|^\kappa \geq 2^\kappa > \kappa$ by Cantor's theorem (proposition 1.5.13). □

This depends critically on the assumption $\Omega = 2$ in the base topos (**Set**); if we drop this assumption it is possible to have small complete proper categories.

+§**1.5.9** Now we shall refute some of the reverse implications. Clearly $\mathrm{R}_0 \not\Rightarrow \mathrm{R}_1$; we now show $\mathrm{R}_2 \not\Rightarrow \mathrm{R}_3$.

Example Consider the cartesian closed category $\mathbf{Pos}_f$ of finite partially ordered sets and monotone functions. Any nonempty object is inhabited and any finite poset with a least element has internal fixpoints, but the singleton is the only reflexive object.

Proof Cartesian closure is well-known, the exponential B^A being just the set of monotone functions from A to B with the pointwise order. Tarski's theorem for finite posets gives the fixpoints: the sequence $\bot, f(\bot), f^2(\bot), f^3(\bot)$,... is monotone and so eventually constant, and the function assigning to f this ultimate value is monotone. Now suppose $A^U \triangleleft U$.

Suppose that $a < b$ is a nontrivial instance of the order relation on A. For $u \in U$ let $f_u : U \to A$ by $f_u(\xi) = a$ if $\xi \leq u$ and b otherwise; f_u and f_v agree on $\{u, v\}$ only if $u \leq v$ and $v \leq u$. This yields a collection of monotone functions $\{f_u : u \in U\}$ with as many elements as U, but it does not exhaust A^U because the constant function with value b does not occur. Hence we have found more elements of A^U than there are in U.

Suppose then that A is discrete, whence so also is A^U. Any map $U \to A$ or $U \to A^U$ is constant on the components of U. So the cardinality of A^U is n^m where n is the cardinality of A and m is the number of components of U. Then the *surjection* forces $n^m \leq m$, which is only possible for $n = 1$. It is, however, possible to have an *inclusion* $A^U \hookrightarrow U$ with $n > 1$. □

+§**1.5.10** $\mathrm{R}_1 \not\Rightarrow \mathrm{R}_2$. For an introduction to K-spaces see Brown [1964].

Example The category of K-spaces and continuous maps is cartesian closed. The unit interval $[0, 1]$ has external but not internal fixpoints.

Proof Let $\mathbb{I} = [0, 1]$ and $\mathbb{R}$ be the reals. $\mathbb{I}$ has external fixpoints by the Intermediate Value Theorem. Suppose $\mathsf{Y} : \mathbb{I}^\mathbb{I} \to \mathbb{I}$ were a (continuous) fixpoint operator. Let $f : \mathbb{R} \to \mathbb{I}^\mathbb{I}$ by

$$f(t, x) \quad = \quad \begin{cases} \mathrm{e}^t x & \text{if } t \leq 0 \\ 1 - \mathrm{e}^{-t}(1-x) & \text{if } t \geq 0 \end{cases}$$

Then $\mathsf{Y}f : \mathbb{R} \to \mathbb{I}$ must take value 0 for $t < 0$ and 1 for $t > 0$ so is not continuous. □

Questions

(a) Does $\mathrm{R}_3 \Rightarrow \mathrm{R}_4$ or $\mathrm{R}_4 \Rightarrow \mathrm{R}_5$?

(b) What is the structure of $\{X \in \mathcal{E} : X \models \mathrm{R}_i\}$?

$^{+}$§**1.5.11** The rest of this section is devoted to showing that certain properties conflict with fixpoints. This is why we have adopted the weaker definition (§1.2.16) for cartesian closure:

Proposition Let $\mathcal{E}$ be lex cartesian closed with every object of $\mathcal{E}$ inhabited. Then $\mathcal{E}$ is degenerate.

Proof Let $A \to 1 \to A$ be the constant endomorphism of an object whose value is the given element. Consider the pair $A \rightrightarrows A$ consisting of this and the identity, and let U be the equaliser of the corresponding elements $1 \rightrightarrows A^A$, so $U \hookrightarrow 1$ is regular mono. But U is inhabited so this is epi and hence iso, which means that the two elements of A^A and so the two maps $A \to A$ are equal, *i.e.* $A \to 1 \to A$ is the identity. Trivially $1 \to A \to 1$ is also the identity, so $A \cong 1$. □

In order to interpret logic, as usually understood, in sets or categories we need to choose some class of inclusions (monos) as the "subsets of elements satisfying some condition". This is called the *Axiom of Comprehension*, and there may be some restrictions on the class of conditions allowed. Since retracts are well behaved, we expect at least these to be admissible as subobjects. However this result together with lemma 1.2.13γ show that we are unable to use intersections to interpret Conjunction.

Another corollary of this is that we cannot solve double fixpoint equations $A \cong fA \cong gA$ for general retracts f, g. We *can* do this if f and g commute, and have done so several times already.

§**1.5.12** Negation and Disjunction also fall. For Negation this is of course the famous *liar paradox*; the complete lattice case is due to Martin Hyland.

Proposition Let A be an object of $\mathcal{E}$ with external fixpoints and which carries a Heyting algebra or complete lattice structure. Then it is degenerate (isomorphic to the terminal object).

Proof Let 0 and 1 be the bottom and top elements w.r.t. the order on A and $\neg : A \to A$ be the "negation" operation. In the Heyting algebra case $\neg a = (a \to 0)$ and in the complete lattice case $\neg a = \bigvee\{b : a \curlywedge b = 0\}$. In either case $\neg 0 = 1$. Now suppose $*$ is a fixpoint of $\neg$. Then in the first case $* = * \curlywedge * = * \curlywedge \neg * = 0$, but then $* = \neg * = \neg 0 = 1$, so $0 = 1$. In the other case $* \curlywedge b = 0$ implies $b \le *$ so $* \curlywedge b = b = 0$; but then $* = \bigvee\{0\} = 0$ and $0 = * = \neg * = \bigvee A = 1$. □

Corollary If every object of $\mathcal{E}$ has external fixpoints and $\mathcal{E}$ has binary coproducts then $\mathcal{E}$ is degenerate.

Proof The coproduct of the terminal object with itself has a Boolean algebra structure, and so is degenerate. It follows that any two global elements $1 \rightrightarrows A$ of an object are equal. So each object has a unique global element, and applying this to the exponentials in $\mathcal{E}$ we deduce that between any two objects there is a unique map. □

§**1.5.13** Now let $\mathcal{E}$ be a topos with subobject classifier (object of truth values) Ω. We shall prove Cantor's theorem.

Lemma If either $A \twoheadrightarrow B$ or $B \hookrightarrow A$ then $\Omega^B \triangleleft \Omega^A$.

Proof The map $\Omega^B \hookrightarrow \Omega^A$ (respectively $\Omega^A \twoheadrightarrow \Omega^B$) given by the exponential contravariant functor gives the inverse image of a subset under the original map. As we shall see in §3.2.16, the inverse image maps have adjoints on both sides, which are quantifiers ($\exists$ on the left, $\forall$ on the right). Either of these provides an inverse, and hence a retract. □

Corollary Suppose $\mathcal{E}$ has an object A with either $A \twoheadrightarrow \Omega^A$ or $\Omega^A \hookrightarrow A$; then it is degenerate.

Proof With $B = \Omega^A$ we then have $\Omega^B \triangleleft B$, so Ω is reflexive. But then it has internal fixpoints, and on the other hand a Heyting algebra structure. □

$^+$§**1.5.14** Here is an idea (from a note dated 24 April 1986) for $[\mathrm{R}_4 \not\Rightarrow \mathrm{R}_5]$. Let Λ be the term $\lambda\beta$-model and $\mathcal{E} = \mathbf{Retr}(\Lambda)$, so by definition $A \models \mathrm{R}_4$ for any object A. Suppose $A \models \mathrm{R}_5$, so $U^U \cong U$ for some object U, say $u : U^U \to U$ and $v = u^{-1}$. Then $u = u\,;v\,;u$, $v = v\,;u\,;v$ and

$$u\,;v = \lambda f.(v\,;u\,;f\,;v\,;u)$$

I conjecture(d) that the only solution to these equations is $u = \mathsf{K}(uu)$, $v = \mathsf{K}u$, *i.e.* $A \cong 1$.

Chapter 2

Categories of Domains

2.1 Inductive Partial Orders

+§**2.1.1** We shall be concerned with models of the λ-calculus which, like $P\omega$, carry an order structure, and in the categories of types arising from them. We begin by repeating the basic order-theoretic result (proposition 1.2.2), and observing its converse.

Proposition Let $(X, \leq)$ be a poset with least element $\perp$ and a least upper bound for any directed set (written $\bigvee^{\uparrow}$). Then any monotone function $f : X \to X$ has a least fixed point.

Proof Put $x_0 = \perp$ and, inductively, $x_\alpha = \bigvee^{\uparrow}\{f(x_\beta) : \beta \in \alpha\}$. (This is a concise form of a transfinite recursive definition; alternatively we put $x_{\alpha+1} = f(x_\alpha)$ for successor ordinals and $x_\lambda = \bigvee^{\uparrow}\{x_\alpha : \alpha < \lambda\}$ for limit ordinals.) f is monotone so the sequence is increasing and the sup is indeed directed. By cardinality (Hartogs' Lemma) the sequence must eventually stabilise, say at x, with $f(x) = x$. If also $y = f(y)$ then $x_\alpha \leq y$ for each α so $x \leq y$. □

+§**2.1.2** Naturally the converse to this involves Choice.

Lemma Let $(X, \leq)$ be a poset, $a \subset X$ a directed subset and κ a cardinal. Suppose X has a least upper bound for any directed set of cardinality less than κ. Then the set of elements lying below the sup of a directed subset of a of cardinality less than κ is an ideal, which we shall call the *κ-completion* of a.

Proof Induction on κ; if κ is a limit cardinal we need only observe that a union of ideals is an ideal, so suppose $\kappa = \lambda^+$. Let b be the λ- and c the κ-completion of a, so b is an ideal. Let $x = \bigvee^{\uparrow} x_\alpha$, $y = \bigvee^{\uparrow} y_\alpha \in c$, expressed as unions of directed subsets of a of size less than κ, so at most λ. Put $z_0 = \perp$ and for $\alpha < \lambda$ let z_α be a bound in b for $\bigvee^{\uparrow}\{z_\beta : \beta < \alpha\}$, x_α and y_α. Then $\{z_\alpha : \alpha < \lambda\}$ is directed of size (at most) λ so has a least upper bound $z \in c$. Also $x_\alpha, y_\alpha \leq z_\alpha \leq z$ so $x, y \leq z$. Hence c is directed, and trivially down-closed. □

Proposition Let $(X, \leq)$ be a poset. Then every monotone function $f : X \to X$ has a least fixed point iff X has a least element ($\perp$) and every directed set has a least upper bound.

Proof The least fixed point of the identity is $\perp$. Let $a \subset X$ be directed of cardinality κ (say $a = \{x_\alpha : \alpha < \kappa\}$) and suppose inductively that X has all directed sups of size $< \kappa$.

Let b be the κ-completion of a and (inductively) for $\alpha \in \kappa$ let $y_\alpha \in b$ be a bound for x_α and $\bigvee^{\uparrow}\{y_\beta : \beta \in \alpha\}$. Now define $f : X \to X$ by

$$f(x) \quad = \quad \begin{cases} x & \text{if } x \text{ is a bound for } a \\ y_\alpha & \text{if } \alpha = \{\beta : y_\beta \leq x\} < \kappa \end{cases}$$

and verify that this is monotone. Observe that $f(x) \leq x$ holds only in the first case, where x is a bound. The fixed points of f are therefore exactly the upper bounds of a, and by hypothesis there is a least fixed point. □

§**2.1.3** In Number Theory the rings in which one can perform Euclid's algorithm for finding the highest common factor are called Euclidean Domains, so since Tarski's theorem characterises posets with $\perp$ and $\bigvee^{\uparrow}$ perhaps *Tarski Domains* would be an appropriate name for them. There seems to be a major industry amongst theoretical computer scientists concerning objects with as little structure as this, and they give them the grandiose title of *complete partial orders* or *cpos*.

The converse we have proved does *not* justify the claim that all directed sups are *necessary* since (i) the proof was nonconstructive (it asked for a fixpoint of a very peculiar function), (ii) we are only interested in *continuous* functions and (iii) for computer science purposes we only want the *effectively given* functions anyway. Nor are the conditions *sufficient*, since the domains for which there is any use are also given in an effective way.

One ought to expect more of something called "complete" than this, and the word is grossly overused anyway. Since Tarski's theorem applies to the semantics of recursion, I shall compromise and call them *inductive partial orders* and write (following Plotkin [1976]) **IPO** for the category having these as objects and as morphisms the functions preserving directed sups.

§**2.1.4** We shall always consider ipos to be provided with the *Scott topology*:

Proposition Let $(X, \leq)$ be a poset with directed sups.

(a) The down-closed subsets of X which are closed under directed sups form the closed sets of a topology.

(b) A function from X to another poset Y with directed sups is continuous w.r.t. this topology iff it preserves directed sups.

Proof Arbitrary unions and intersections of down-closed sets are down-closed. If we also ask for closure under directed sups have only *finite* unions. If $f : X \to Y$ preserves directed sups (whence by considering the sup of any two comparable elements it's monotone) then its inverse image preserves closure. Conversely if f is continuous and $a \subset X$ directed consider the closure of the image of a, which is $\downarrow(\bigvee^{\uparrow}(fa))$; the inverse image of this is closed and so contains $\bigvee^{\uparrow} a$. □

However Johnstone [1979] has shown that the Scott topology need not be sober, although as soon as the conditions which we shall regard as minimal (*viz.* continuity, §2.3.5) are imposed upon the poset then it is always sober.

§**2.1.5** The above derives *topology* from *order*. Conversely any topological space carries a preorder on its points, the *specialisation order*. Here $x \leq y$ if every open set which contains x also contains y.

Lemma The specialisation order $\leq$ on a space X is

(a) a partial order (antisymmetric) iff X is T_0,

(b) discrete iff X is T_1. □

Topology can say nothing about spaces below T_0, so we shall always assume this holds. [However Fred Linton has pointed out to me that in a *finite* T_0 space, $\mathsf{int}\,\bar{a} \subset \overline{\mathsf{int}\,a}$ (the interior of the closure is in the closure of the interior) for any subset a, so that there are *fewer* than fourteen sets obtainable using closure and complement, whereas there is a nine-point example of a non-T_0 space with fourteen.] Since Hausdorff $= \mathrm{T}_2 \Rightarrow \mathrm{T}_1$, part (b) explains why this order does not feature in geometric or analytic topology.

Lemma

(c) Let $(X, \leq)$ be a poset with directed sups, and σ its Scott topology. Then the specialisation order on (X, σ) is $\leq$.

(d) Let (X, τ) be a sober space and $\leq$ its specialisation order. Then $(X, \leq)$ has directed sups and any τ-open set is Scott-open. □

Of course this order structure applies not only to the points of a space but also to continuous maps between them.

Lemma

(e) Let X, Y be spaces and $f, g : X \rightrightarrows Y$ continuous maps between them. Then $\forall x. fx \leq gx$ in the specialisation order on the points of Y iff $\forall V. f^{-1}V \subset g^{-1}V$ in the inclusion order on open subsets of X. □

§**2.1.6** Now we consider products and exponentials in **IPO**.

Proposition

(a) **IPO** has arbitrary products, created by the forgetful functor **IPO** $\to$ **Set**.

(b) A function of several (perhaps infinitely many) arguments is jointly continuous iff it is separately continuous.

(c) the Scott and Tychonov topologies on a (possibly infinite) product agree. □

Proof

[a] We calculate $\bigvee^{\uparrow}$ componentwise.

[b] By proposition 2.1.4b we have to preserve $\bigvee^{\uparrow}$; but this is calculated componentwise so we only need preserve it componentwise.

[c] Let $X_i \in$ **IPO** for $i \in I$. Write S, T for the product sets with the respective topologies. By definition $S \to T$ is continuous; we have to show that if $U \subset \prod_i X_i$ is Scott-open then it is Tychonov-open. Let $(x_i : i \in I) \in U$; then for $a \in P_f(I)$ put

$$x_i^a \quad = \quad \begin{cases} x_i \text{if } i \in a \\ \perp \text{otherwise} \end{cases}$$

so $\bigvee^{\uparrow}\{(x_i^a : i \in I) : a \in P_f(I)\} = (x_i)$. But U is Scott-open so $(x_i^a : i \in I) \in U$ for some $a = \{i_1, ..., i_r\}$. Let $U_1, ..., U_r$ be the projections of U onto the corresponding factors of the product; then the rectangle $U_1 \times ... \times U_r \times \prod_{i \notin a} X_i$ is a basic Tychonov-open set containing $(x_i : i \in I)$ and lying within U. □

This good behaviour for products does not apply to all limits. Though **IPO** does have cofiltered limits, **IPO** $\to$ **Sp** does not preserve them.

Example Let C_n be the "flat domain" with 2^n (maximal) points and $\perp$. Consider the diagram with vertices the C_n and surjections $C_{n+1} \to C_n$ which identify pairs of maximal points in the obvious way. The limit C_∞ of this diagram in **Sp** is obtained from the Cantor set by adding a closed point $\perp$ whose only neighbourhood is the whole space. The limit in **IPO** is the same set, but with the discrete topology on the maximal points (*cf.* example 5.2.6d). □

§**2.1.7**

Lemma In the set of Scott-continuous functions between two ipos with the pointwise order, sups and infs (where they exist) are calculated pointwise. In particular this poset is an ipo. □

Proposition IPO is cartesian closed and has internal fixpoints.

Proof The ipo constructed in the lemma is the exponential. We found the fixed points in proposition 1.2.1, and this construction is readily seen to be continuous. □

We shall shortly strengthen this reflexivity property (R_2 from §1.5.1) to R_5. If we were to drop the requirement that there be a least element we would get *all* limits but of course lose fixpoints.

$^+$§**2.1.8** When *can* we take pullbacks in **IPO**? A map in a category against which we can take the pullback with any map with the same codomain is said to be *carrable* (French, means *squarable*).

Proposition

(a) A Scott continuous map $p : X \to Y$ between ipos is carrable iff it is a projection (§1.2.4).

(b) The pullback of a carrable map is carrable.

(c) The composite of two carrable maps is carrable.

(d) Any terminal projection is carrable.

Proof

[a,⇒] If $p : X \to Y$ is carrable let $y \in Y$ and take the pullback against the corresponding $\ulcorner y \urcorner : 1 \hookrightarrow Y$. This is the inverse image of y in X and has to have a least element, which we call iy. We now have to show that $iy \le x$ iff $y \le px$; but since $y = p(iy)$, "only if" is trivial. Let $y \le px$; consider the map $2 \to Y$ by $0 \mapsto y$, $1 \mapsto px$ and let W be the pullback. $(0, iy)$ is the least element of W, and in particular it is less than $(1, x)$, so $iy \le x$. p therefore has a left adjoint monotone function, and this then has to preserve *all*, not just directed, sups.

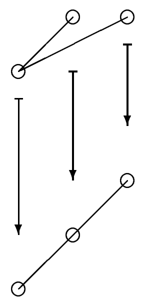
$$\begin{tikzcd} W \arrow[r] \arrow[d, shift left=1.5] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, shift left=1.5, "p"] \\ Z \arrow[u, dotted, shift left=1.5] \arrow[u, phantom, "\dashv"] \arrow[r, "f"] & Y \arrow[u, shift left=1.5, "i"] \arrow[u, phantom, "\dashv"] \end{tikzcd}$$

[⇐] Let $p : X \to Y$ be a projection and $f : Z \to Y$ any continuous function. Let $W = X \times_Y Z$ be the pullback in **Set**, which consists of the set of pairs (x, z) with $p(x) = f(z)$. With the componentwise order this is a poset with directed sups. $(i(f\perp_Z), \perp_Z)$ is the least element.

(b,c) are standard properties of pullbacks (1.2.14).

(d) Terminal maps are projections because of $\perp$; they are carrable because we have products. □

Example The following map shows that the fourth sentence of the proof is not redundant: each point of the codomain has a least preimage but the map is not a projection.



This result will become relevant when we discuss fibrations of ipos in chapter V. Parts (b) to (d) are the axioms for a class of display maps (§4.3.2). Adjoint pairs of maps like these will play a major rôle in the rest of the work.

§**2.1.9** As already remarked in §2.1.3, **IPO** is not particularly interesting in itself but serves as a convenient category in which to perform basic constructions. In fact we shall take more interest in certain cartesian closed full subcategories of it.

Lemma Let $\mathcal{C}$ be a full subcategory of **IPO** which is cartesian closed *in itself*. Then the products and exponentials in $\mathcal{C}$ are those from **IPO**.

Proof

[1] Let T be the terminal, and X any other object of $\mathcal{C}$. By definition there is a unique $\mathcal{C}$-map from X to T, and hence by fullness a unique **IPO**-map. But there is a map $\mathsf{K}t$ for each $t \in T$, so T has just one point.

[$\times$] Let $A, B \in \mathcal{C}$; write $A \times B$ for their product in **IPO** and $A \otimes B$ for that in $\mathcal{C}$. We have to show that the comparison map $A \otimes B \to A \times B$ in **IPO** is bijective and order-reflecting. Each element of $A \otimes B$ (map $T \to A \otimes B$) gives a pair $\langle a, b\rangle$ (with $a \in A$, $b \in B$) by composition with the $\mathcal{C}$-product projections, and conversely by the universality of $\otimes$ each such pair arises from a unique $[a, b] \in A \otimes B$ in this way. Hence we have bijectivity. For order-reflection, let $\langle a, b\rangle \leq \langle a', b'\rangle$ in $A \times B$; by transitivity we need only consider the (similar) cases $a \leq a'$, $b = b'$ and $a = a'$, $b \leq b'$. We have a (unique) map $A \to A \otimes B$ from $1_A : A \to A$ and $\mathsf{K}b : A \to B$; by monotonicity, $a \leq a'$ implies $[a, b] \leq [a, b']$.

[$\to$] Let $A, B \in \mathcal{C}$. By the exponential adjunctions and fullness there is an natural bijection among elements of the $\mathcal{C}$-exponential, continuous functions $A \to B$ and elements of the **IPO**-exponential. To show that this is an order isomorphism, let $C \in \mathcal{C}$ be an arbitrary nontrivial domain and $\perp \neq 1 \in C$ (since $2 \lhd C$, *cf.* Lemma 2.3.1b); given $f \leq g$, we can extend these to $h : C \times A \to B$ with $h(\perp, a) = f(a)$ and $h(1, a) = g(a)$. □

§**2.1.10** We devote the remainder of this section to a discussion of finite spaces which will be needed in section 6.

Proposition There is an equivalence between

(α) Finite T_0 spaces and continuous maps

(β) Finite posets and monotone maps

(γ) Finite distributive lattices and opposite homomorphisms.

Proof [$\alpha \Rightarrow \beta$] By the specialisation order. [$\beta \Rightarrow \alpha$] There is a unique (T_0) topology on a finite poset for which the specialisation order is the given relation. [$\alpha \Rightarrow \gamma$] The topology. [$\beta \Rightarrow \gamma$] The lattice of upper sets. [$\gamma \Rightarrow \beta$] The subposet of coprime elements. □

More generally, the prime filters on an (infinite) distributive lattice provide a space called the *spectrum*, whose compact open sets are the elements of the given lattice. A space which arises in this way is said to be *coherent* (see §2.4.6).

+§**2.1.11** First we characterise finite surjections and their splittings.

Definition

(a) A subset S of a poset X is *convex* if when $s_1 \leq x \leq s_2$ for $s_1, s_2 \in S$ and $x \in X$ then $x \in S$.

(b) A subset of a space is called *locally closed* if it is the intersection of an open set with a closed set.

(c) A collection of subsets $X_1, ..., X_n$ of a poset X is *mutually convex* if each is convex and if $x_{i_1} \leq y_{i_2}$, $x_{i_2} \leq y_{i_2}$, ... $x_{i_{k-1}} \leq y_{i_k}$, $x_k \leq y_1$, for some $i_1, \ldots i_k$, $x_{i_j}, y_{i_j} \in X_{i_j}$, then $i_1 = i_2 = \cdots = i_k$. [This definition was incorrectly stated (with $k = 2$) in the original.]

Lemma Let X be any space and $S \subset X$.

(a) $S = f^{-1}(y)$ for some continuous $f : X \to Y$ with Y T_0 finite and $y \in Y$ iff S is locally closed in X.

(b) Such a set is convex in the specialisation order. [This is used in §2.6.2.]

(c) Let X be an ipo; then S is locally closed iff it is convex and closed under and inaccessible by directed sup (*i.e.* if $s \leq \bigvee^{\uparrow} x_i$ then $\exists i, s'.s' \leq x_i$.)

(d) Let X be an algebraic poset (2.3.4); then S is locally closed iff $S = \{\bigvee^{\uparrow} D : D \subset C\}$ where $C = S \cap X_{fp}$.

(e) In (d), any convex $C \subset X_{fp}$ arises in this way.

Proof

[a,⇒] Put $U = f^{-1}(\uparrow y)$ and $A = f^{-1}(\downarrow y)$.

[⇐] Let $u, v : X \to 2$ be the characteristic functions of the open sets U and $X \backslash A$, and put $f = \langle u, v\rangle : X \to 2^2 = Y$, $y = \langle 1, 0\rangle$.

[b] Suppose $s_1 \leq x \leq s_2$ with $f(s_1) = f(s_2) = y$. Then $y \leq f(x) \leq y$.

[c,⇒] (a) applied to Scott-continuity.

[⇐] $\uparrow S$ is open and $\downarrow S$ closed in the Scott topology.

[d,e] follow from (c). □

Proposition Let $X = X_0 \cup ... \cup X_{n-1}$ be a partition of a space.

(a) This arises from a surjection to a finite T_0 space iff the sets are locally closed and mutually convex.

(b) It arises from a retract with image $x_i \in X_i$ iff each X_i is locally closed and whenever $u \in X_i$, $v \leq X_j$ with $u \leq v$ in the specialisation order we have $x_i \leq x_j$.

Proof Necessity in each case is trivial.

[a] By mutual convexity we may suppose (since there are only finitely many of them) that the X_j are numbered in such a way that if $u \in X_i$, $v \in X_j$, $u \leq v$ then $i \leq j$. Let $\mathbf{n} = \{0, 1, ..., n-1\}$ with $0 < 1 < ... < n-1$ and $f : X \to \mathbf{n}$ by $X_i \to \{i\}$. This is monotone, surjective and gives rise to the partition: it remains to show that it is continuous. Put $U_i = \bigcup\{X_j : i \leq j\}$. Suppose $X_i = V_i \cap A_i$ and $v \in V$; by the partition $v \in X_j$, say, so by the numbering $i \leq j$, and $v \in U_i$; hence $V_i \subset U_i$. Conversely let $v \in U_i$, so $v \in X_j$ for some $i \leq j$ and then $v \in V_j \subset U_j \subset U_i$. Hence $U_i = \bigcup\{V_j : i \leq j\}$ and is open.

[b] Mutual convexity is forced by the additional condition. Let Y be the poset with the same elements $0, 1, ..., n-1$ as before, but only those instances of the order relation which are forced by monotonicity. Then $Y \cong \{x_0, ..., x_{n-1}\} \subset X$. □

${}^{+}$§**2.1.12** The corresponding problem for injections is much harder. The following result simply amplifies the T_0 separation axiom.

Lemma Let X be a T_0 space and $S \subset X$ nonempty finite. Then there is a finite T_0 space T and a continuous map $X \to T$ such that $S \hookrightarrow X \to T$ is continuous and bijective (though not necessarily a homeomorphism).

Proof Suppose $S = \{s_0, ..., s_{n-1}\}$ is enumerated in such a way that $s_i \leq s_j$ implies $i \leq j$ (*sic*), using T_0. By the definition of the specialisation order, for $i < j$ we have some open V_{ij} with $s_i \notin V_{ij}$ and $s_j \in V_{ij}$. Put

$$U_i = \bigcup_k \{\bigcap_j \{V_{jk} : j < k\} : i \leq k\} \quad \text{and} \quad A_i = X \backslash U_{i+1}$$

Then $X = U_0 \supset U_1 \supset ... \supset U_{n-1}$, $s_i \in U_i \cap A_i$ and these sets partition X. Put $T = \mathbf{n} = \{0, 1, ..., n-1\}$ and define $X \to T$ by $U_i \cap A_i \to \{i\}$. This is continuous because the inverse image of $\uparrow i$ is U_i; $S \to X \to T$ is clearly bijective. We can cut down the order relation on T in the same way as remarked in the proof of proposition 2.1.11b. There are obvious counterexamples to homeomorphism. □

In order to split an injection we need to choose a partition satisfying the conditions of proposition 2.1.11b. To construct this it would be useful to know, for instance, that $\downarrow\uparrow x$ is always closed. However this is not true in general (example 2.5.7b). In fact in section 6 we shall retreat to a "lattice-like" case.

The proof of lemma 2.6.3a is a variant of this.

2.2 The Limit Colimit Coincidence

§**2.2.1** In this section we begin to see the real power of (suitable subcategories of) **IPO**. Tarski's theorem provides a very simple semantics for recursion, but we find that similar results apply to the *category* as well as its objects. In other words we can solve *recursive domain equations* such as that $\Lambda \cong \Lambda^\Lambda$ which defines an η-algebra using fixpoint techniques. At the foundation of this is the *limit-colimit coincidence* first observed by Scott [1972] in the context of complete lattices; the technique was subsequently developed by Smyth and Plotkin [1978]. As we work through this theory the similarities between domains and categories of domains become more and more conspicuous, and in many places we either deliberately confuse, or are in danger of accidentally confusing, object and meta-levels. Eventually (and this is the major aim of this work) we find that we can push this as far as the construction of a "type of types".

§**2.2.2** Perhaps Scott's "D_∞" model of the λ-calculus is the best place to start. We want a "fixed point" of the construction $X \mapsto X^X$. There are three rather obvious problems here. First, after Cantor's theorem (1.5.13) we might think that X^X is rather a lot bigger than X. Second, "fixed point" means that $f(X)$ is *equal* to X, and it is the very first lesson of Category Theory that equality of objects just does not arise naturally in Mathematics. We shall return to this notion in §5.3.6. Finally, this construction is not functorial, because $(X, Y) \mapsto Y^X$ is contravariant in its first (raised) argument.

The first problem really just shows that our intuition (based on *discrete* sets) is simply wrong. Of course any nontrivial Λ with $\Lambda \cong \Lambda^\Lambda$ is infinite, but its infinity is of a "fluid" rather than "granular" kind.

There are two different solutions to the second problem. One involves working in a combinatory model such as $P\omega$, where the types have some kind of canonical representation as elements (though there may still be distinct isomorphic types). Whilst this is very pleasant to work with, in the absence of explanation it's just conjuring.

The other solution, naturally offered by Category Theory, is to replace *equality* by *isomorphism*. We also learn from this discipline that isomorphism is not a *property* of a pair of objects but a *structure* between them: we have to quote the maps explicitly and carry them around with us in the construction. More generally, the categorical analogue of Tarski's theorem requires us to name comparison maps between the objects in the diagram, where previously we replied on the mere *existence* of (an instance of) the partial order relation.

The third problem we also sweep under the carpet (in the first instance) by a trick, namely to move to an auxiliary class of morphisms. In chapter V we attempt to explain what this is all about, although really there the "explanation" amounts only to a comparison with the $\mathbf{Retr}(P\omega)$ case.

§**2.2.3** Scott's D_∞ construction of an η-model in **IPO** takes us from the weakest to the strongest of the reflexivity properties listed in §1.5.1. We start with some "seed" (such as the two-element lattice 2) and repeatedly apply the function space construction until it stabilises (recall the notation X_m from §1.2.17). From lemma 1.5.4 it is *necessary* that the seed be inhabited, and this turns out to be *sufficient* because we use the point to give a particular retraction $X \lhd X^X$ (by dropping a variable).

In the context of categories rather than posets we need to specify the comparison maps in the diagram. In our case the first of these is the given retraction $X \lhd X^X = X_1$, and subsequent ones are obtained by applying the function-space functor to this. This is *not* the same as dropping a variable between X_m and X_{m+1}.

Whereas with Tarski's theorem the order relation (and $\perp$) forced the terms in the sequence to be increasing, in the categorical case we have to supply comparison maps. [No: the same argument still applies; there is a unique map $\perp \to F\perp$ since $\perp$ is initial, and applying F provides the other finite terms. The colimiting cocone gives maps to vertices at limit ordinals; then applying F again to the whole diagram we have another cocone, to which there is a unique mediating map from the colimit.]

Thus we need a *pointed endofunctor*, *i.e.* a functor $F : \mathcal{C} \to \mathcal{C}$ together with a natural transformation $\eta_X : X \to FX$. This may be iterated in a diagram of the form

$$X \xrightarrow{\eta_X} FX \xrightarrow{F\eta_X} F^2X \xrightarrow{F^2\eta_X} F^3X \xrightarrow{F^3\eta_X} \cdots F^mX \xrightarrow{F^m\eta_X} F^{m+1}X \cdots$$

The reason for using $F^m(\eta_X)$ rather than $\eta_{(F^mX)}$ is that when we apply F to the whole diagram we then get another copy of it; hence if F preserves the (filtered) colimit then this colimit, $F^\infty X$, will be a fixed point of F in the sense that $F(F^\infty X) \cong F^\infty X$.

But we have already remarked that the function-space construction is not functorial, so what use is this to our case? Well, we change category. In stead of using continuous maps in **IPO** we use retractions; then in applying the function space construction, which is partly covariant and partly contravariant, we use whichever of the two maps comes to hand. Thus given $(i,p) : X \lhd Y$ we define $(i^p, p^i) : X^X \lhd Y^Y$ by $i^p f = p\,;f\,;i$ and $p^i g = i\,;g\,;p$.

$^+$§**2.2.4** Let us briefly consider the general retract case.

Let $\mathcal{C}$ be any category. Write $\mathcal{C}^{\mathrm{re}}$ for the category whose objects are those of $\mathcal{C}$ and whose morphisms $X \to Y$ are pairs (i,p) where $i : X \hookrightarrow Y$ and $p : Y \twoheadrightarrow X$ satisfy $i\,;p = 1_X$. Let $X : I \to \mathcal{C}^{\mathrm{re}}$ be any diagram. Write $\mathsf{lim}\, X_i$ for the limit of the diagram in $\mathcal{C}$ whose arrows are the epi parts, $\mathsf{colim}\, X_i$ for the colimit of the diagram of monos in $\mathcal{C}$ and $\mathsf{bilim}\, X_i$ for the limit in $\mathcal{C}^{\mathrm{re}}$, where these exist.

Lemma

(a) If lim and bilim both exist then they are isomorphic.

(b) Likewise colim and bilim.

(c) If lim and colim both exist and are isomorphic then so does bilim (and is isomorphic to them).

Proof

[a] Let $\pi_i : \mathsf{lim} \to X_i$ and $(i_i, p_i) : X_i \to \mathsf{bilim}$ be the limiting cones. By the universal property of lim there is a comparison map $p : \mathsf{bilim} \to \mathsf{lim}$ in $\mathcal{C}$ with $p\,;\pi_i = p_i$. Put $\iota_i = i_i\,; p$, then $\iota_i\,;\pi_i = 1$ so we have a cone in $\mathcal{C}^{\mathrm{re}}$. But bilim is universal, so we have a comparison map $(\iota, \pi) : \mathsf{bilim} \to \mathsf{lim}$ in $\mathcal{C}^{\mathrm{re}}$ with $\iota\,;\pi = 1$, $i_i\,;\iota = \iota_i$ and $\pi\,;p_i = \pi_i$. Now $\iota\,;\pi_i = \iota\,;\pi\,;p_i = p_i$ so ι satisfies the universal property of p, whence $\iota = p$. Also $\pi;\iota;\pi_i = \pi;\iota;\pi;p_i = \pi;p_i = \pi_i = 1;\pi_i$ so again by the definition of lim, $\pi\,;\iota = 1$. Hence $\mathsf{lim} \cong \mathsf{bilim}$. [**Why does** $i_i\,;p_i = \mathsf{id}$**?**]

[b] The same argument in $\mathcal{C}^{op}$.

[c] Put $\mathsf{bilim} = \mathsf{lim} = \mathsf{colim}$ with limiting cones $i_i : X_i \to \mathsf{colim}$ and $p_i : \mathsf{lim} \to X_i$. Then $(i_i, p_i) : \mathsf{bilim} \to X_i$ is limiting. □

If lim, colim and bilim all exist for some diagram of retracts we say $\mathcal{C}$ *has the limit-colimit coincidence* for that diagram. As defined the existence of bilim does not imply that of lim or colim (even when the other exists), but by convention we shall only speak of a bilimit when all three exist.

Example $\mathbf{ContLat}^{em}$ has coproducts, calculated as products in **ContLat**, but **ContLat** does not have coproducts. □

+§**2.2.5** What is the effect of this in a cartesian closed category? Let us recall the effect of the $\times$ and $\to$ functors in $\mathcal{C}^{\mathrm{re}}$. Given $(i, p) : X' \lhd X$ and $(j, q) : Y' \lhd Y$ we have $(i \times j, p \times q) : X' \times Y' \lhd X \times Y$ and $(j^p, i^q) : {Y'}^{X'} \lhd Y^X$, where $j^p(f) = p\,; f\,; j$.

Proposition Let $\mathcal{C}$ be a cartesian closed category with the limit colimit coincidence for the four diagrams $X : I \to \mathcal{C}^{\mathrm{re}}$, $Y : J \to \mathcal{C}^{\mathrm{re}}$, $X \times Y : I \times J \to \mathcal{C}^{\mathrm{re}}$ and $[X \to Y] : I \times J \to \mathcal{C}^{\mathrm{re}}$. Then $\mathsf{bilim}(X_i \times Y_j) \cong (\mathsf{bilim}\, X_i) \times (\mathsf{bilim}\, Y_j)$ and $\mathsf{bilim}(X_i \to Y_j) \cong (\mathsf{bilim}\, X_i) \to (\mathsf{bilim}\, Y_j)$.

Proof $\times$ commutes with all limits anyway, so $\mathsf{lim}(X_i \times Y_j) \cong (\mathsf{lim}\, X_i) \times (\mathsf{lim}\, Y_j)$ and by the lemma the result follows. In particular $W \times -$ preserves bilimits (of course if it has a right adjoint then it preserves colimits).

For exponentials,

$$\begin{array}{rll}
& \mathsf{hom}_{\mathcal{C}}(W \times \mathsf{bilim}\, X_i, \mathsf{bilim}\, Y_j) & \\
\cong & \mathsf{hom}_{\mathcal{C}}(\mathsf{bilim}(W \times X_i), \mathsf{bilim}\, Y_j) & \text{above} \\
\cong & \mathsf{hom}_{\mathcal{C}}(\mathsf{colim}(W \times X_i), \mathsf{lim}\, Y_j) & \text{hypothesis} \\
\cong & \mathsf{lim}_i\, \mathsf{lim}_j\, \mathsf{hom}_{\mathcal{C}}(W \times X_i, Y_j) & \text{definition of lim and colim} \\
\cong & \mathsf{lim}_i\, \mathsf{lim}_j\, \mathsf{hom}_{\mathcal{C}}(W, {Y_j}^{X_i}) & \text{definition of } {Y_j}^{X_i} \\
\cong & \mathsf{hom}_{\mathcal{C}}(W, \mathsf{lim}_{i,j}\, {Y_j}^{X_i}) & \text{definition of lim} \\
\cong & \mathsf{hom}_{\mathcal{C}}(W, \mathsf{bilim}_{i,j}\, {Y_j}^{X_i}) & \text{hypothesis}
\end{array}$$

□

§**2.2.6** The retraction $X \lhd X^X$ given by dropping a variable at $\perp$ in **IPO** is a coclosure. We are interested only in this case. Of course this depends on the 2-structure of **IPO**.

Proposition **IPO** has the limit-colimit coincidence for filtered diagrams of embeddings.

Proof Let $X : I \to \mathbf{IPO}^{em}$ be filtered. The *limit* $\mathsf{lim}\, X_i$ consists of I-indexed families (x_i) with $x_i \in X_i$ which are *compatible* in the sense that the *projection* $X_i \to X_j$ takes x_i to x_j. These families are ordered pointwise. This has directed sups by continuity, and has bottom ($\perp$) since projections preserve it.

The *colimit poset* is to be found embedded in $\lim X_i$ as follows. Given $x \in X_i$ we define $x_j \in X_j$ for any $j \in J$ for which there is an arrow $i \to j$ as the image of x under the appropriate embedding (if there are two such then by filteredness they are coequalised by a further embedding, but this is mono so they were equal to start with); for other j we take by filteredness a suitable bound k for i and j and project from k to j (going to a further bound gives the same answer because the composite of an embedding with its projection is the identity).

However $(x_i) \in \lim X_i$ lies above, and indeed is the directed sup of, the embeddings of its components $\{x_i : i \in I\}$ so $\lim X_i$ is also the colimit ipo. $\square$

§**2.2.7** Write $\mathbf{IPO}^{em}$ and $\mathbf{IPO}^{\mathrm{pr}}$ for the categories of ipos and respectively embeddings and projections, so $\mathbf{IPO}^{em} \simeq (\mathbf{IPO}^{\mathrm{pr}})^{op}$.

Proposition Any functor $F : \mathbf{IPO}^n \times (\mathbf{IPO}^{op})^m \to \mathbf{IPO}$ which preserves the order between continuous functions also preserves coclosures, splittings thereof, embeddings and projections (but interchanges the last two in the contravariant case). It therefore restricts to a functor $(\mathbf{IPO}^{\mathrm{pr}})^{n+m} \to \mathbf{IPO}^{\mathrm{pr}}$. $\square$

We use this to get round the contravariance of the exponential functor in its first argument. It will turn out, however, that the significance of embeddings and projections in **IPO** is far greater than this mere trick.

Since clearly $\mathbf{IPO}^{em}$ has an initial object (the singleton) and these filtered colimits, it bears a striking resemblance to its own objects. The difference, of course, is that it is a *large category* whilst they are *small posets*. In order, therefore, to take full advantage of the coincidence of *structure* we would have to move to **ICat**, the (2-)category of small inductive categories (with initial object and all filtered colimits) and Scott-continuous (*i.e.* filtered-colimit preserving) functors. Unfortunately, space forbids discussion of this extension of the theory.

There remains, however, an important distinction of *size* which we cannot abolish and have to circumvent by surgery.

We shall, however, take immediate advantage of the parallel by calling a functor *continuous* if it preserves bilimits, *i.e.* filtered colimits in $\mathbf{IPO}^{em}$. From proposition 2.2.5, $\times$ and $\to$ are continuous.

Warning The term “continuous” in standard category theory means preservation of all *limits*. Of course “complete” and “continuous” followed “limit” from General Topology.

§**2.2.8** We have now collected the machinery to solve domain equations such as $D \cong D^D$ which amount to finding a fixed point of an continuous endofunctor of $\mathbf{IPO}^{em}$.

Proposition

(a) Let F be a continuous endofunctor of $\mathbf{IPO}^{em}$; then F has a fixed point.

(b) Every object of **IPO** is the image of some coclosure on an η-model.

Proof

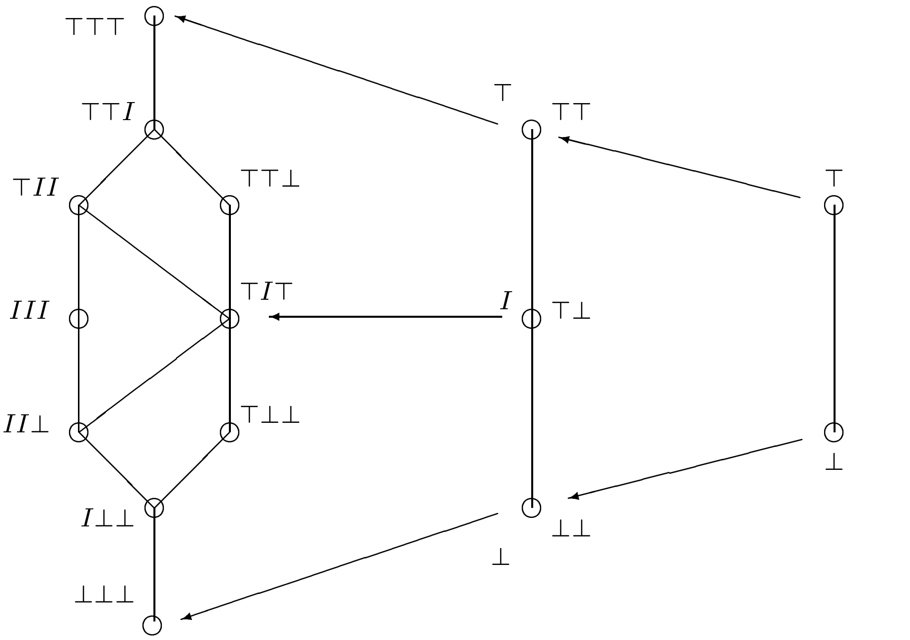
[a] We construct the diagram as in §2.2.3 with X the singleton (initial object of $\mathbf{IPO}^{em}$) and $X \to FX$ the unique map. Then $F^\infty X$ is a fixed point.

[b] In the case of the function space construction we get nothing interesting from the singleton. However starting from any $D \in \mathbf{IPO}$ we have $D \lhd D_1 \lhd \ldots \lhd D_\infty$ where D_∞ is an η-model. $\square$

§**2.2.9** Recall the notion of head normal form from §1.1.13; the (infinite) tree of terms we obtain by developing the recursive head normal form so far and substituting $\perp$ for subterms (where one term lies above another in the tree if it is obtained by filling in some terms for occurrences of $\perp$) we call the *Böhm tree*. This gives a directed set of finite *syntactic* approximants to a term, each of which is in normal form in $\Lambda[\perp, x_1, ...]$. We say that a model Λ in **IPO** is *continuous* (§1.4.9) if the interpretation of a term is the directed sup of the interpretation of its Böhm approximants.

In the D_∞ model we calculate the interpretation of a term as the directed sup of its approximants in D_n for $n \geq n_0$, where n_0 measures the complexity of its functionality in the obvious way. These approximants give the same values as the Böhm approximants (it is crucial to start by dropping a variable in order to obtain this: Park [1976]), and so D_∞ is continuous. In particular Y is the *least* fixpoint combinator because the Böhm approximants of $\mathsf{Y}f$ are just $f^n\perp$.

The D_∞ construction is illustrated by the figure; observe that the embedding of D_1 in D_2 is *not* by dropping a variable.



$$\begin{tikzcd} D_2 = (2^2)^{(2^2)} \arrow[r, shift right, "\mathsf{pre}(\mathsf{K})\ ;\ \mathsf{post}(\mathsf{ev}_\perp)"'] & D_1 = 2^2 \arrow[l, shift right, "\mathsf{pre}(\mathsf{ev}_\perp)\ ;\ \mathsf{post}(\mathsf{K})"'] \arrow[r, shift right, "\mathsf{ev}"'] & D_0 = 2 \arrow[l, shift right, "\mathsf{K}"'] \end{tikzcd}$$

Figure 2.2.9: Construction of D_∞ (after Stoy)

§**2.2.10** Moving to the auxiliary class of morphisms was only needed in this technique because of our wish to treat the function space construction. We can do the same for any pointed endofunctor on a category.

Definition Let $\eta : 1 \to F : \mathcal{C} \to \mathcal{C}$ be a pointed endofunctor. An *algebra* for (η, F) is an object $X \in \mathcal{C}$ together with a *structure map* $\xi : FX \to X$ such that $\eta_X\ ;\ \xi = 1_X$. A *homomorphism* of F-algebras is a $\mathcal{C}$-map $f : X \to Y$ making the square commute:

$$\begin{tikzcd} FX \arrow[r, "Ff"] \arrow[d, "\xi"'] & FY \arrow[d, "\upsilon"] \\ X \arrow[r, "f"] & Y \end{tikzcd}$$

We therefore have a category F-**Alg** and a forgetful functor $F\text{-}\mathbf{Alg} \to \mathcal{C}$.

Lemma If (X, ξ) is initial in F-**Alg** then ξ is an isomorphism, indeed it is the initial fixed point of F. □

This reduces the problem of finding fixed points of F to that of finding initial F-algebras, or more generally to finding a *free algebra functor*, *i.e.* a left adjoint $\mathcal{C} \to F\text{-}\mathbf{Alg}$ to the forgetful functor.

Proposition If $\mathcal{C}$ has and F preserves countable filtered colimits then there is a free algebra functor; in this case F has fixed points.

Proof Same technique as in proposition 2.2.8a. If also $Y \cong FY$ then the unique map $X \to Y$ extends to a cone over the diagram (using $F^m X \to F^m Y \cong Y$). The mediating map $F^\infty X \to Y$ is the unique such map, so that $F^\infty X$ is the initial algebra. □

In Universal Algebra, preserving (countable) filtered colimits corresponds to *finitary* algebraic theories. This is a very common and important phenomenon in Computer Science, since we frequently encounter complicated data types such as stacks, trees, *etc.*, which are specified by means of algebraic operations without recursion.

$^+$§**2.2.11** Without the hypothesis of continuity we need to construct a transfinite diagram, and the category may not possess large filtered colimits. There is, for instance, no free complete Boolean algebra on countably many generators.

On the other hand it is not impossible for a proper category (*i.e.* not a poset) to have *all* (large) filtered colimits. Indeed any finite category in which idempotents split, and more generally any bilimit of such categories (see §2.5.6), has this property. Hence the generalisation from **IPO** to **ICat** is not a trivial one.

We cannot afford the space to prove the following generalisation of Tarski's theorem.

Fact Let $\mathcal{C}$ be a small category with all filtered colimits.

(a) Let $\eta : 1 \to F : \mathcal{C} \to \mathcal{C}$ be a pointed endofunctor of $\mathcal{C}$ (not necessarily continuous). Then there is a free F-algebra functor.

(b) An arbitrary intersection of (replete) reflective subcategories of $\mathcal{C}$ is reflective. □

$^+$§**2.2.12** Embeddings and projections of ipos will play a major rôle in the theory. It will, however, be useful (in §5.2) to make a generalisation. Recall that these two maps are adjoint and their composite is the identity. It seems not generally known that the latter condition is not necessary for most occurrences of these maps in the theory, and apart from this ignorance I have seen no argument to motivate it.

Definition Let $f : X \to Y$ be continuous, *i.e.* in **IPO**. Then f is

(a) a *homomorphism* if it has a left adjoint

(b) a *comparison* if it has a continuous right adjoint

(c) a *projection* if it is a surjective homomorphism (Notation $\twoheadrightarrow$)

(d) an *embedding* if it is an injective comparison (Notation $\rightarrowtail$).

[POSTSCRIPT: The corresponding notions in stable domain theory have shown a very close connection between (rigid) comparisons and the (Berry) order relation, namely that stable functions have a factorisation system whose intermediate object is the trace, whilst instances of the order relation give rise to rigid comparisons between the traces. The right adjoint (rigid homomorphism) preserves any internal structure of the domains which is preserved by binary meets bounded above (pullbacks) — in particular if the latter distribute over arbitrary joins then the homomorphism itself has a right adjoint. Rigid homomorphisms therefore behave like logical functors between topos,s locally cartesian closed categories, *etc.* The name, however, was generalised from continuous lattices (Carl Gunter used the word in the same way and, I believe, for similar reasons), but the "algebraic" notions suggested by this are, for general domains, so weak as to be misleading. I now feel that the choice of the word homomorphism for a Scott-continuous functor with a left adjoint between arbitrary ipos was mistaken.]

Write $\mathbf{IPO}^{hm}$, $\mathbf{IPO}^{cp}$, $\mathbf{IPO}^{\mathrm{pr}}$ and $\mathbf{IPO}^{em}$ for the four categories; there is also a duality at the 2-level, so

$$\mathbf{IPO}^{em}(X,Y) \quad\simeq\quad \mathbf{IPO}^{\mathrm{pr}}(Y,X)^{op} \quad\subset\quad \mathbf{IPO}^{cp}(X,Y) \quad\simeq\quad \mathbf{IPO}^{hm}(Y,X)^{op}$$

Propositions 2.2.5-8 generalise immediately from projections to homomorphisms (or from embeddings to comparisons), although of course lemma 2.1.8 does not. Any covariant or contravariant functor of however many variables which preserves order restricts to a covariant functor on $\mathcal{C}^{cp}$, and many important functors (notably products and function-spaces) are continuous in the sense of preserving bilimits. In particular we prove

Lemma $\mathbf{IPO}^{cp}$ has filtered colimits.

Proof Let $X : I \to \mathbf{IPO}^{cp}$ be filtered. If $i \to j$ in I, the composite of the two maps gives rise to a closure operator on X_i, and the system of closure operators corresponding to all of the points beyond $i \in I$ is directed. Let c_i be the sup of these closure operators on X_i and Y_i its image. If $i \to i'$ in I we have a comparison $X_i \to X_{i'}$, and this gives rise to an *embedding* $Y_i \rightarrowtail Y_{i'}$. We therefore have a diagram $Y : I \to \mathbf{IPO}^{em}$. This has a colimit by proposition 2.2.5, and this is the colimit of the original diagram. □

A better proof of the general limit colimit coincidence will be found in *Homomorphisms, bilimits and saturated domains.*

$^{+}$§**2.2.13**

Note Some of the results claimed in the original version of the remainder of this section were fallacious and have been removed. A more thorough treatment of limits and colimits in categories of domains was undertaken in my manuscript *Homomorphisms, Bilimits and Saturated Domains,* to which the reader is referred.

We can take the pullback of two homomorphisms in **IPO** (cf proposition 2.1.8a), but the result is not a pullback in $\mathbf{IPO}^{hm}$ because the mediating maps from other diagrams do not have left adjoints (though they may occasionally have *right* adjoints). Indeed

Lemma Let $X \in \mathbf{IPO}$. Then the diagonal $X \to X \times X$ has a left adjoint iff X is a complete lattice.

Proof It is binary join. □

Example The pullback of a homomorphism against another in **IPO** need not be a homomorphism.

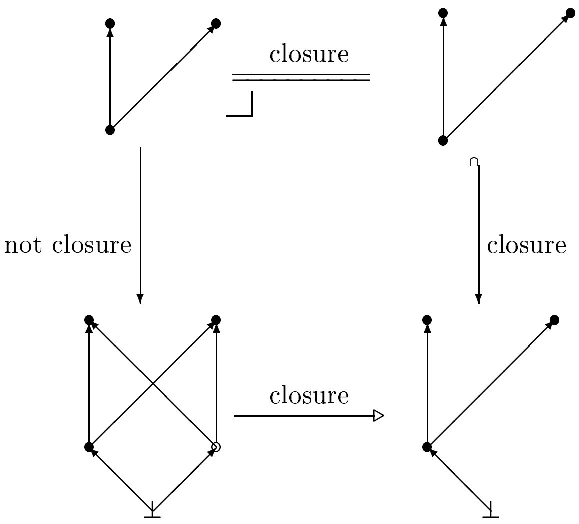


Figure 2.2.13: Pullback of homomorphisms

$^+$§**2.2.14** It is a standard result in elementary category theory that pullbacks and a terminal object suffice to construct all finite limits in a category, and that if further we have cofiltered limits then we have all limits. Unfortunately this does not work in the case of projections, because the pullback does not exist as a functor. Even to show that the maps in the limiting cone are homomorphisms, as we have done for pullbacks, we would need the stronger result.

Example Let X be an ipo but not a lattice. The limit of each of the diagrams in the figure is X and the diagonal, $\Delta : X \to X \times X$, which is not a homomorphism, is a map in the limiting cone. The last diagram is of the form $\mathcal{T}^2 \to \mathbf{IPO}$.

A category I is called a *tree* it is non-empty and for any two points $i, j \in I$ there is a *unique* sequence of non-identity arrows in alternate directions between them.

Theorem Let $d : I \to \mathbf{IPO}^{hm}$ be a tree-shaped diagram of projections. Then the maps in the limiting cone are projections.

Proof We use pullbacks to construct limits of finite trees; and arbitrary tree is the filtered union of its finite subtrees. The limit is therefore a bilimit of pullbacks. □

This result is proved in greater detail in [Taylor 1987].

$^+$§**2.2.15**

Lemma

(a)
$$\mathbf{IPO}^{cp}(X,2)^{op} \cong \mathbf{IPO}^{hm}(2,X) \cong \begin{cases} X \text{if } X \text{ has } \top \\ \emptyset \text{ otherwise} \end{cases}$$

where the comparison corresponding to $x \in X$, which we write as $[x]$, is the characteristic function of $X \setminus \downarrow x$, and the homomorphism takes $\bot$ and $\top$ to x and $\top$ respectively.

(b)
$$\mathbf{IPO}^{hm}(X,2)^{op} \cong \mathbf{IPO}^{cp}(2,X) \cong X_{fp}$$

where the homomorphism corresponding to $x \in X_{fp}$ is the characteristic function of $\uparrow x$ and the comparison takes $\bot$ and $\top$ to $\bot$ and x respectively. □

$$\begin{tikzcd} X \times X \arrow[r, shift left, "\pi_0"] \arrow[r, shift right, "\pi_1"'] & X \end{tikzcd}
\qquad
\begin{tikzcd} X \times X \arrow[d, "\pi_0"'] \arrow[dr] & X \arrow[d, "\mathrm{id}"] \arrow[dl] \\ X & X \end{tikzcd}$$

$$\begin{tikzcd} X \arrow[drr] \arrow[dr] & X \times X \arrow[dl] \arrow[dr] & X \arrow[dll] \arrow[dl] \\ X & X & X \end{tikzcd}
\qquad
\begin{tikzcd} X \times X \arrow[r, "\pi_0"] \arrow[d, "\pi_1"'] & X \arrow[d] & X \arrow[l] \arrow[d] \\ X \arrow[r] & 1 & X \arrow[l] \\ X \arrow[r] \arrow[u] & X \arrow[u] & X \arrow[l] \arrow[u] \end{tikzcd}$$

Figure 2.2.14: Diagrams equivalent to equalisers

2.3 Continuous Lattices and Posets

§**2.3.1** As already remarked, we shall be interested in the category **IPO** itself only as a convenient place in which to do calculations; the categories which are of concern to us will be full subcategories closed under products, exponentials, retracts and countable bilimits. This section is chiefly concerned with the smallest such category, $\mathbf{ContLat}_\omega$, which is in fact equivalent to $\mathbf{Retr}(P\omega)$. Wherever possible, however, our arguments will be given in a form applicable to all continuous *posets*, not just *lattices*.

What is the smallest nontrivial category of (countably based algebraic) domains? It has a nonsingleton object D, so this has two distinct points $\bot$ and u, with $\bot \le u$. This is known as the *Sierpinski space*, written 2, and it has the following property.

Definition An *injective* space I is one such that given any diagram of spaces and continuous maps

$$\begin{tikzcd} U \arrow[r, hook] \arrow[d] & X \arrow[dl, dotted] \\ I \end{tikzcd}$$

where $U \hookrightarrow X$ is a subspace inclusion (*i.e.* a subset of U is open iff it is the intersection of U with an open subset of X, which is equivalent to saying that the map is a regular mono in **Sp**) there is some (not necessarily unique) map $X \to I$ making the triangle commute.

Lemma The Sierpinski space 2 is injective and is a retract of any space which is T_0 but not T_1.

Proof

[a] The inverse image of the open point of 2 under the continuous map is an open subset V of U, and there is a largest open set W in X with $W \cap U = V$. Let the extension take the larger value on W and the smaller outside it.

[b] Let $x < y$ in the specialisation order on X. Then $2 \cong \{x, y\} \subset X$ so by injectivity we have a postinverse. □

§**2.3.2** The importance of the Sierpinski space to topology really lies in the fact that it *classifies open sets*, *i.e.* for any $X \in \mathbf{Sp}$ there is a (natural, order-preserving) bijection between open sets of X and continuous functions $X \to 2$. This makes it a *cogenerator* for $\mathrm{T}_0\mathbf{Sp}$, *i.e.* given $f, g : X \rightrightarrows Y$ if whenever we take the composites with maps $Y \to 2$ they are equal then the original maps were already equal. (This is the dual notion to 1.2.10.)

Proposition A T_0 space is injective iff it is a retract of a Tychonov power of 2.

Proof Powers of injectives (specifically 2) are injective by taking the (canonical) extension on each factor; for retracts first make an extension (of the composite with the mono) into the large space then use the epi to bring it back.

Let X be injective with κ open sets. Let $X \hookrightarrow 2^\kappa$ by taking each point to the collection of open sets containing it. Verify that this is a subspace inclusion (it is mono by T_0). By injectivity the identity on X extends to a retraction. □

Observe that 2^X is the Scott topology on an ipo X.

§**2.3.3** The next task is to characterise those posets which occur as retracts of full powerset lattices (powers of the Sierpinski space). We shall do this in four stages. First we shall characterise images of *closure* operators on 2^κ (which give algebraic lattices). Next we introduce continuous posets by the "way below" relation and show that they occur as images of *coclosures* of algebraic posets. Then we show that continuous lattices are the algebras for directed sups and continuous arbitrary infs. Finally we show that retracts of continuous posets are also continuous.

Definition Let X be an ipo and x an element of it. We say x is *compact* or *finite* in X if whenever $x \leq \bigvee^{\uparrow} a$, then already $x \leq y$ for some $y \in a$. Write X_{fp} for the set of compact elements of X. (The reason for this notation is that the categorical analogue is called "finitely presentable".) Notice we always have $\perp \in X_{fp}$.

Example $x \in 2^\kappa$ is compact iff it is finite *quâ* subset of κ. □

Proposition Any comparison of ipos (Definition 2.2.12b) preserves compactness.

Proof Let $f : X \to Y$ have a continuous right adjoint $g : Y \to X$, let $x \in X$ be compact and $b \subset Y$ directed with $fx \leq \bigvee^{\uparrow} b$ in Y. Then by the adjunction and continuity of g, $x \leq g(\bigvee^{\uparrow} b) = \bigvee^{\uparrow}(gb)$ in X. Hence by compactness $x \leq gy$ for some $y \in b$ and again by adjointness $fx \leq y$. □

§**2.3.4**

Definition An element x of an ipo X is *finitely approximable* if it satisfies $x = \bigvee^{\uparrow}\{k \in X_{fp} : k \leq x\}$. Note that it is part of the condition that this set be directed. X is *algebraic* if every element is finitely approximable.

Algebraic *lattices* occur as the subobject lattices of finitary algebraic theories, *e.g.* groups, rings, modules, lattices but not fields (not algebraic) or complete lattices (not finitary). Any subgroup of a group, for instance, is the directed union of the finitely generated (*not* finite) subgroups it contains. Closure operators have also been familiar in mathematics for a long time in similar contexts (group generated by a subset, for instance).

Proposition The images of continuous closure operators on 2^κ are precisely the algebraic lattices.

Proof Observe first that 2^κ is itself algebraic.

[⇒] Let $c^2 = c \geq 1$ be a continuous closure operator on an algebraic poset X with image Y and let $y \in Y$. Then $y = \bigvee^{\uparrow} a$ where $a = \{k \in X_{fp} : k \leq x\}$. But for $k \in a$, ck is compact in X by proposition 2.3.3, and $k \leq ck \leq y$. Hence $b = \{ck : k \in a\} \subset \{l \in Y_{fp} : l \leq y\}$ is directed and $x = \bigvee^{\uparrow} b = \bigvee^{\uparrow}\{l \in Y_{fp} : l \leq y\}$ as required.

[$\Leftarrow$] Let X be algebraic and $\kappa = |X_{fp}|$. Define $X \hookrightarrow 2^\kappa$ by $x \mapsto \{k \in X_{fp} : k \le x\}$. Then X is the image of the closure operator $c(a) = \{k \in X_{fp} : k \le \bigvee a\}$. $\square$

§**2.3.5** We generalise algebraic to continuous posets by making the notion of finiteness or compactness a relative one.

Definition Let X be an ipo. For $x, y \in X$ we say x is *way below* y if whenever $y \le \bigvee^{\uparrow} a$ for $a \subset X$ directed then already $x \le z$ for some $z \in a$. Write $x \ll y$ for this relation and $\downarrow\!\!\downarrow y = \{x \in X : x \ll y\}$, $\Uparrow x = \{y \in X : x \ll y\}$. $x \in X$ is *approximable* if $x = \bigvee^{\uparrow} \downarrow\!\!\downarrow x$ (directedness being again part of the condition), and X is *continuous* if every element is approximable.

In the case of the lattice of open sets of the reals, $\mathbb{R}$, $x \ll y$ means that there is a compact set lying between x and y. A space is *locally compact* iff its lattice of open sets is continuous; $\mathbb{R}$, for instance, has this property. Scott [1972] gave an argument that continuous posets are the appropriate notion of approximate computation, although most of his followers have since retreated to the algebraic condition.

Proposition

(a) If $x \ll y$ then $x \le y$

(b) If $x' \le x \ll y \le y'$ then $x' \ll y'$

(c) $x \ll x$ iff $x \in X_{fp}$

(d) Comparisons preserve $\ll$.

Proof [a-c] Easy. [d] As proposition 2.3.3. $\square$

§**2.3.6** There's a canonical way of embedding a continuous poset in an algebraic one.

Proposition Let X be a continuous ipo and $\mathsf{Idl}X$ its poset of ideals. Then $\mathsf{Idl}X$ is an algebraic ipo on which there is a continuous coclosure of which X is the image. Specifically the embedding is $x \mapsto \downarrow\!\!\downarrow x$ and the the projection $a \mapsto \bigvee^{\uparrow} a$.

Proof The compact elements of $\mathsf{Idl}X$ are the principal ideals, $\{\downarrow x : x \in X\}$. This is the *largest* ideal whose sup is x, so $\downarrow$ is right adjoint to $\bigvee^{\uparrow}$ (but is not continuous). $\downarrow\!\!\downarrow x$, on the other hand, is the *smallest* ideal with sup x, so $\downarrow\!\!\downarrow$ is left adjoint. $\square$

For categories there is a construction called Ind which generalises Idl in the sense that it freely adjoins filtered colimits. A category $\mathcal{C}$ *has* filtered colimits iff the inclusion $\mathcal{C} \hookrightarrow \mathsf{Ind}\,\mathcal{C}$ has a left adjoint. A *continuous category* is then one for which this functor has a further left adjoint. Johnstone and Joyal [1982] have used these to solve the analoguous questions for toposes which are addressed in this section for spaces.

$^-$§**2.3.7** We shall now give the algebraic characterisation of continuous lattices, which shows again the importance of adjoint pairs of continuous maps.

Lemma In a continuous lattice inf distributes over $\bigvee^{\uparrow}$.

Proof Let X be continuous and $\{x_{jk} : j \in J, k \in K(j)\}$ be a family of elements of X such that $\{x_{jk} : k \in K(j)\}$ is directed for each $j \in J$. Let M be the set of functions $f : J \to \bigcup_{j \in J} K(j)$ with $\forall j. fj \in K(j)$. Then clearly

$$l = \bigvee^{\uparrow}_{f \in M} \mathsf{inf}_{j \in J}\, x_{j,f(j)} \quad \le \quad \mathsf{inf}_{j \in J} \bigvee^{\uparrow}_{k \in K(j)} x_{jk} = r$$

so we have to show the reverse inequality. By continuity it suffices to show that if $u \ll r$ then $u \leq l$. By the definition of (inf and) $\ll$ this means $\forall j.\exists k \in K(j).u \leq x_{jk}$, so choose $f \in M$ with $\forall j.u \leq x_{j,f(j)}$; but then $u \leq l$. □

Proposition Continuous lattices are exactly the algebras for directed sup and continuous arbitrary inf; the homomorphisms are continuous functions which have left adjoints.

Proof The lemma provides one implication, and since a monotone function between complete lattices has a left adjoint iff it preserves arbitrary infs it is clear what the homomorphisms are. Hence it remains to show that such an algebra is continuous, for which we have only to show that for each element there is a least ideal whose sup is that element. But by the Lemma the family of ideals with this property is closed under intersection. □

Write $\mathbf{ContLat}^{hm}$ for the category of continuous lattices (*i.e.* $(\mathsf{inf}, \bigvee^{\uparrow})$-algebras) and homomorphisms. Since in general the regular epis in a category of algebras are the surjective homomorphisms, in this case they are the projections.

The free functor $\mathbf{Set} \to \mathbf{ContLat}^{hm}$ was identified by Day [1975]; it yields the lattice of filters on the powerset. Of more interest to us is the free functor from **IPO**; this gives the lattice of open filters, as it does on **Sp**. We shall prove the corresponding result for **bcCont** in proposition 2.5.9.

This result is the reason for the term *homomorphism*; the term *comparison* will be explained in §5.1.10.

$^{-}$§**2.3.8** We now complete the characterisation of injectives.

Lemma A retract of a continuous poset is continuous.

Proof Let $f = f^2 : X \to X$ in **IPO** with image Y. We first show that if $x \in X$, $y \in Y$ with $x \ll y$ in X then $fx \ll y$ in Y. For if $y \leq \bigvee^{\uparrow}_Y b = \bigvee^{\uparrow}_X b$ for directed $b \subset Y$ then $x \leq y'$ for some $y' \in b$ and so $fx \leq fy' = y$ in Y. Hence for $y \in Y$ we have

$$y = fy = f(\bigvee_X^{\uparrow}\{x \in X : x \ll_X y\}) = \bigvee_Y^{\uparrow}\{fx : x \ll_X y\} \leq \bigvee_Y^{\uparrow} \downarrow\!\!\downarrow y$$

as required, checking that the sets are indeed directed. □

We may sum these and some other results up as follows.

Theorem The following are equivalent for an T_0 space X with its specialisation order (and the Scott topology by default):

(α) X is injective

(β) X is a continuous lattice

(γ) X is a retract of a (Tychonov) power of the Sierpiński space

(δ) X has arbitrary infs which distribute over directed sups

(ϵ) X is an algebra for the filter monad.

For distributive X we may add:

(ζ) X is the open set lattice of an *exponentiable* space Y, *i.e.* one such that the functor $- \times Y :$ $\mathbf{Sp} \to \mathbf{Sp}$ has a right adjoint. □

§**2.3.9** In order to describe precisely the Scott topology on a continuous poset we need the *interpolation property* of $\ll$.

Lemma Let X be a continuous poset and $x \ll z$ in X. Then for some y, $x \ll y \ll z$.

Proof Consider the set $S = \{v \in X : (\exists y \in X)(v \ll y \ll z)\}$. We first show that S is an ideal; it is clearly nonempty and down-closed. If $v_1, v_2 \in S$ then $v_i \ll y_i \ll z$ for some $y_1, y_2 \in X$. Then since $\downdownarrows z$ is directed we can find $y \ll z$ with $y_1 \leq y \geq y_2$, and then since $\downdownarrows y$ is directed we can find $z \ll y$ (whence $z \in S$) with $z_1 \leq z \geq z_2$. Also, we clearly have $\bigvee^{\uparrow} S = \bigvee^{\uparrow}\{\bigvee^{\uparrow} \downdownarrows y : y \ll z\} = \bigvee^{\uparrow}\{y : y \ll z\} = z$. But by hypothesis $x \ll z$ so $x \in S$. $\square$

Proposition Let X be a continuous poset.

(a) the sets $\Uparrow x$ for $x \in X$ form a base for the Scott topology.

(b) the compact open sets of X are precisely those of the form $\Uparrow\{k_1, ..., k_n\} = \Uparrow k_1 \cup ... \cup \Uparrow k_n$ for $k_1, ..., k_n \in X_{fp}$.

Proof

[a] If $\bigvee^{\uparrow} D \in \Uparrow x$ then by the lemma $x \ll d \ll \bigvee^{\uparrow} D$ for some $d \in D$, so $\Uparrow x$ is open. Conversely let U be open (and hence upper) and $u \in U$; then $\downdownarrows u \cap U \neq \emptyset$, so $u \in \Uparrow x$ for some $x \in U$, and $\Uparrow x \subset U$.

[b] The given sets are compact since they have a finite set of minimal elements. Let $x \in X$ with $\Uparrow x$ compact. Then there are $y_1, ..., y_r \in \Uparrow x$ with $\Uparrow x = \Uparrow y_1 \cup ... \cup \Uparrow y_r$; w.l.o.g. this set is minimal. But by the lemma $x \ll z \ll y_1$, so $y_i \ll z \ll y_1$ for some i, and $i = 1$ by minimality, whence $y_1 \in X_{fp}$. Likewise for $y_2, ..., y_r$, so $\Uparrow x$ is of the given form. Since $\{\Uparrow x : x \in X\}$ is a basis, the result follows. $\square$

We may characterise the Scott topology on continuous and algebraic posets: see Johnstone [1983] VII theorem 2.6(iii) and corollary 2.9.

Proposition

(c) The Scott topology on a continuous poset is a completely distributive lattice, and every such arises in this way.

(d) Further the lattice is algebraic iff the poset is. $\square$

§**2.3.10** For the purpose of doing detailed calculations with continuous ipos it is useful to have a small class of easily describable functions of which any other can be expressed as a directed sup. These are called *step functions*.

Lemma Let X, Y be continuous ipos and $x \in X$, $y \in Y$. Define

$$[x \Rightarrow y] \quad : \quad x' \quad \mapsto \quad \begin{cases} y \text{ if } x \ll x' \text{ in } X \\ \bot \text{otherwise} \end{cases}$$

Then $[x \Rightarrow y]$ is continuous, and its dependence on x is contravariant and on y is covariant. It is compact in $X \to Y$ iff x and y are compact in X, Y resp.

Proof This is really the characteristic function of the open set $\Uparrow x$. $\square$

Proposition ContLat, **AlgLat**, **ContLat**$_\omega$ and **AlgLat**$_\omega$ are cartesian closed.

Proof We have only to show that the function space $X \to Y$ as calculated in **IPO** inherits the properties of X and Y. Since it has arbitrary infs constructed pointwise, it is clearly a lattice. Let $f : X \to Y$ be continuous. Observe that for $f : X \to Y$, if $y \ll fx$ in y then $[x \Rightarrow y] \ll f$ in $X \to Y$; indeed f is the directed sup of all such functions. In the algebraic case we may choose x, y to be compact. If X and Y have countable bases then the step functions provide a countable basis for $X \to Y$. $\square$

ContPos and **AlgPos** are not, however, closed under exponentiation (2.4.11-13).

$^+$§**2.3.11** Finally we turn to the limit-colimit coincidence for **ContPos**.

Lemma Let $X : I \to \mathbf{ContPos}^{cp}$ be a filtered diagram of comparisons of continuous posets; then the bilimit is continuous. If the posets in the diagram are lattices or algebraic then so (respectively) is the bilimit; if they are countably based and I is countable then the bilimit is also countably based.

Proof Recall first that $\ll$ is preserved by comparisons. Let X_∞ be the bilimit and $x \in X_\infty$ have images $x_i \in X_i$ under the projections. Then $x_i = \bigvee^{\uparrow}\{y_i \in X_i : y_i \ll x_i\}$ in each X_i. But identifying the x_i and y_i with their images under embedding, $x = \bigvee^{\uparrow}\{x_i : i \in I\} = \bigvee^{\uparrow}\{\bigvee^{\uparrow}\{y_i \in X_i : y_i \ll x_i\} : i \in I\} = \bigvee^{\uparrow}\{y_i : y_i \ll x_i, i \in I\}$, whence x is approximated. Replacing $\ll$ by compactness, the same argument applies to the algebraic case. Since projections preserve infs, we may calculate these componentwise and so being a lattice is preserved. In the countable case we take the union of the bases (identified under embedding).

Theorem **ContLat**, **AlgLat**, $\mathbf{ContLat}_\omega$ and $\mathbf{AlgLat}_\omega$ are full subcategories of **IPO** closed under products, exponentials, images of closures and countable bilimits. Moreover they are the smallest nontrivial such categories also closed under respectively retracts and small bimilits, small bilimits, retracts, and nothing else. □

These properties essentially amount to what we shall define as a category of domains. However we want to make a generalisation first, and that will be the subject of the next section.

2.4 Bifinite Posets

$^+$§**2.4.1** When Scott first provided mathematical structures for the semantics of programming languages, he believed that the irrelevance of *top* to computation was outweighed by the mathematical usefulness of lattices. This view quickly lost favour, chiefly because Plotkin's [1976] powerdomain construction (which is intended as a semantics for nondeterminism, but is outside the scope of this work) does not preserve the property of being a lattice. Indeed the domains in which Scott himself subsequently took an interest (which sometimes have his name attached to them, though we write $\mathbf{bcAlg}_\omega$) are themselves continuous (or algebraic) lattices minus their top elements.

There is a larger class of algebraic posets which *is* closed under the powerdomain constructions, whilst still also being closed under exponentiation. Indeed Smyth [1982] showed that it is the largest cartesian closed full subcategory of countably based algebraic posets.

Definition A *bifinite poset* is one which may be expressed as a bilimit of finite posets. Write $\mathbf{BiPos}_\mathrm{f}$ for the category of bifinite posets and Scott-continuous maps, and $\mathsf{Bi}_\omega\mathbf{Pos}_f$ for the full subcategory of countably based such.

(Space forbids discussion of the prefix Bi for general 2-categories.)

We may imagine that bifinite posets describe the limit of a sequence of finite computers. Calculations with integers on finite machines may give *overflow*, but if they dont overflow then they are correct. An arbitrary computation with integers may be performed on a sequence of machines, each twice as big and twice as fast as the one before. Where two machines both succeed they necessarily agree, and there is a two-way comparison between any machine and any bigger one. This comparison, in information terms, must be an embedding-projection pair.

We shall *not* settle upon this or an other particular class of ipos to call domains. The reason for this is that there is in fact significant interest in the variability, for the following reason. In §3.3 we shall give a brief introduction to classifying toposes; for present purposes one of these is a (generalised) space whose points are the models of a particular theory in a certain fragment of logic (namely geometric logic). From this, spaces correspond to theories and classes of spaces to classes of theories. It turns out, though again discussion of this goes beyond the intentions of this work, that certain well-known categories of domains correspond to certain identifiable fragments of logic.

We shall derive from this a *hierarchy* of categories of domains, each one in various flavours according whether they are countably based and whether they are closed under retracts. We shall explore this hierarchy a little in the next section, and in the final section show that we may essentially always assume that we are working with a category of the form $\mathbf{Retr}(\Lambda)$. This section is concerned with the top of that hierarchy, the bifinite posets.

$^+$§**2.4.2** We begin with a version of Plotkin's characterisation, of which the essential idea is the

Definition Let X be a poset.

(a) $m \in X$ is a *minimal upper bound* or *mub* of $S \subset X$ if $S \leq m$ (*i.e.* $\forall s \in S.s \leq m$) and if $S \leq m' \leq m$ then $m' = m$.

(b) $S \subset X$ has a *complete set of mubs* if for any $S \leq x \in X$ there's some mub m of S with $m \leq x$.

(c) X is *mub-complete* is any $S \subset X$ has a complete set of mubs.

(d) $S \subset X$ is *mub-closed* if any *subset* of S has a complete set of mubs which lie within S.

Clearly any subset of a mub-complete poset has a mub-closure, obtained by successively adjoining mubs.

Lemma Let X be a poset and $S \subset X$ finite. Then S is the image of a (unique) coclosure on X iff it is mub-closed and $S \subset X_{fp}$.

Proof

[$\Rightarrow$] Suppose $p : X \to S$ is a continuous right adjoint to the inclusion $i : S \rightarrowtail X$. The elements of S are compact in S because the latter is finite, and embeddings preserve compactness (proposition 2.3.3), so $S \subset X_{fp}$. Let m be a mub of $A \subset S$, so $A \leq m$; then since p is a projections fixing A, $A = pA \leq pm \leq m$; but m is minimal above A so $pm = m$, *i.e.* $m \in S$. Suppose $x \geq A \subset S$; then $A = pA \leq px \leq x$; choose $m \in S$ minimal with $A \leq m \leq px$; repeating the argument shows that m is a mub in X.

[$\Leftarrow$] Suppose $S \subset X_{fp}$ is mub closed and let $x \in X$. Consider the set $S \cap \downarrow x$, which is by hypothesis finite. Let px be a mub for it below x; by mub-closure $px \in S$ so is the greatest element of $S \cap \downarrow x$. This gives a right adjoint p to i. It is continuous because px is compact. □

Example Let X be a complete lattice (or at least a boundedly complete poset). Then the finite coclosures of X are given by the finite sub-$\curlyvee$-semilattices of X_{fp}. □

Notation We shall adopt the convention of using "curly" symbols $\curlyvee$ and $\curlywedge$ for operations within a lattice to avoid confusion with the logical connectives.

$^+$§**2.4.3** Plotkin's characterisation of bifinite posets proceeded by explicitly adjoining mubs. We may simplify this to

Proposition For a poset X tfae:

(α) X is bifinite

(β) $\mathsf{id} : X \to X$ is the directed sup of the continuous coclosures on X with finite image.

(γ) X is algebraic and every finite set of compact elements is contained in a finite mub closed set of compact elements.

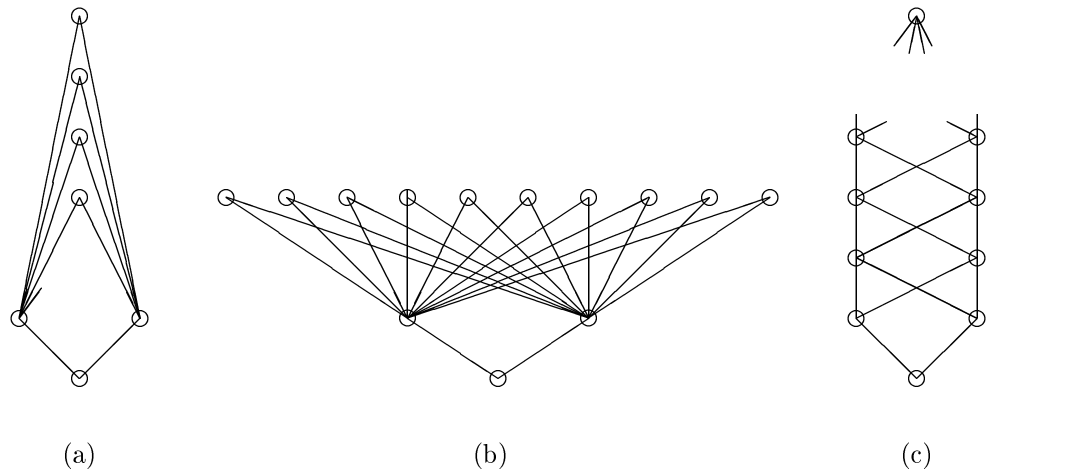


Figure 2.4.4: Three non-bifinite domains

Proof

[$\alpha \Rightarrow \beta$] Let $X = \mathsf{bilim}\, X_i$, so $X \to X_i$ are coclosures with finite image. Let $x \in X$ with projections x_i. Then $x = \bigvee^{\uparrow} x_i$.

[$\beta \Rightarrow \gamma$] For $x \in X$, $x_i \in X_{fp} \cap \downarrow x$ and $x = \bigvee^{\uparrow} x_i$, so X is algebraic. Let $k \in X_{fp}$; then $k = \bigvee^{\uparrow} k_i$, so $k = k_i \in X_i$ for some i. For finite $S \subset X_{fp}$ choose X_i by the directedness of I; this is a finite mub-closed set by lemma 2.4.2.

[$\gamma \Rightarrow \alpha$] Let $(X_i : i \in I)$ be the diagram of finite mub-closed subsets of X under inclusion. This is directed and the inclusions have continuous right adjoints. The colimit of these posets accounts for all of the compact elements of X, so $X = \mathsf{bilim}\, X_i$. □

Example A complete lattice or boundedly complete poset is bifinite iff it is algebraic. □

Question In (γ) does it suffice for pairs?

Answer No. Figure 1.7 from [Jung 1987] provides a counterexample.

§**2.4.4** We record here for future use the three essential examples of algebraic ipos which are *not* bifinite.

Examples

(a) Some bounded pair of compact elements of X fails to have a mub below a given bound, so X is not mub-complete; also $X \to X$ is not continuous.

(b) X is mub-complete but some pair has infinitely many mubs.

(c) Any finite set of compact elements of X has a finite complete set of mubs, but the construction of adjoining mubs continues indefinitely; again $X \to X$ is not continuous.

§**2.4.5** We have already done the work in constructing products and exponentials of bifinite posets. $\mathbf{Pos}_f$ has all finite limits, colimits and exponentials (§1.5.9), and the full subcategory of posets with bottom is still cartesian closed.

Proposition $\mathbf{BiPos}_\mathrm{f}$ has the limit-colimit coincidence and is cartesian closed. Also

$$\begin{aligned}(\mathsf{bilim}\, X_i) \times (\mathsf{bilim}\, Y_j) &\cong \mathsf{bilim}(X_i \times Y_j)\\ (\mathsf{bilim}\, X_i) \to (\mathsf{bilim}\, Y_j) &\cong \mathsf{bilim}(X_i \to Y_j)\end{aligned}$$

Proof The diagram defining the bilimit is the filtered union of those defining the terms. From §2.2.5, the product and exponential functors are continuous. Alternatively, let $S \subset X$, $T \subset Y$ be mub-closed sets in bifinite posets. Then $S \times T$ and $S \to T$ are mub-closed sets in $X \times Y$ and $X \to Y$ respectively. In the first case we take the obvious componentwise projections. In the second case the projection of $f : X \to Y$ is the composite $S \rightarrowtail X \to Y \to T$. The compact functions are therefore those which reduce X to a finite set and apply some function into a finite subset of Y. □

+§**2.4.6** Recall that the Scott topology on an algebraic poset is based by the sets $\uparrow k = \{x \in X : k \leq x\}$ for compact k. These are of course *compact* open sets. A general compact open set is of the form $\uparrow a = \{x \in X : k \leq x \text{ for some } k \in a\}$ where a is a finite (possibly empty) set of finite elements.

Definition A space is *coherent* it it is sober and compact, it has a base of compact open sets, and the intersection of any two compact open sets is again compact.

The specialisation order on a coherent space gives a *profinite poset*, *i.e.* one which is the limit of a diagram of finite posets and monotone maps (*not necessarily projections*). Since I believe full coherent logic will turn out to be needed in Computer Science in due course, we are not at liberty to use the term *profinite* for *bifinite*.

Lemma In a coherent space codirected infs exist.

Proof We may calculate them in the finite projections. □

[It was claimed that they distribute over directed sup. This is nonsense.]

Example 2.1.6 shows that the coherent topology on a profinite poset need not be the Scott topology.

Proposition A continuous poset is coherent iff it is algebraic and any finite set of compact elements has a complete finite set of minimal upper bounds.

Proof We use the characterisation of compact opens in an continuous poset X (proposition 2.3.9b). Let $S \subset X_{fp}$ be finite; then $U = \{x \in X : \forall s \in S. s \leq x\}$ is an intersection of compact opens.

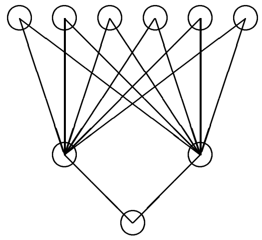
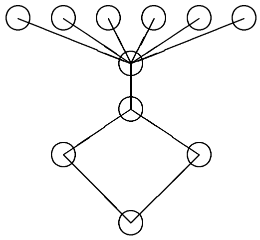
[⇒] U is compact open and so of the form $\uparrow T$ with $T \subset X_{fp}$ finite. Taking a minimal such form we have a complete finite set of mubs for S.

[⇐] $U = \uparrow T$, where T is the (complete finite) set of mubs for S. This is compact open. □

Notice that we haven't used the full force of the characterisation of bifinite posets, so the "radio mast" (Example 2.4.4c) is coherent algebraic but not bifinite.

§**2.4.7** The characterisation of §2.4.3 applies equally well if we drop the requirement that our posets have bottom elements. Unlike ipos we do *not* gain all finite limits.

Example The diagram on the left is not bifinite, but it is the equaliser in **IPO** of two endofunctions of the diagram on the right, *viz.* the two functions which are the identity everywhere apart from the two ponts in the middle, and send both of those points to one of them. (Carl Gunter)

Instead of having a unique bottom element we have a smallest (finite) mub-closed set called the *root*, which contains the minimal elements. An abstract root is a finite poset with no nontrivial coclosure operator. We say that a continuous functor F *preserves* a root R if R is the root of $F(R)$. This terminology is due to Gunter [1985].

Proposition A continuous functor F has a bifinite fixpoint with root R iff F preserves R.

Proof

[$\Rightarrow$] Let $D \cong F(D)$ be a fixpoint with root R. Then R is the image of a coclosure on D and so since F is continuous $F(R)$ is a coclosure of $F(D)$. But R is the *least* coclosure on D so factors through this one and so R is the root of $F(R)$.

[$\Leftarrow$] Let $i : R \rightarrowtail F(R)$ be an embedding. Form a diagram from $F^n(i) : F^n(R) \rightarrowtail F^{n+1}(R)$ and take take its bilimit D as before. Since F is continuous $D \cong F(D)$ with root R. □

Example The function space construction preserves only the singleton root.

Proof Clearly otherwise we could embed any bifinite poset in an η-model and hence get least fixed points. More explicitly, if m is minimal in D then $\mathsf{K}m$ is minimal in D^D, but on the other hand so is the least coclosure (onto the root). Hence unless D has a (unique) bottom, D^D has more minimal elements and so a bigger root than D. □

From corollary 2.4.12 in fact any β-model in **AlgPos** has $\perp$.

$^+$**§2.4.8** With a great deal of grubbing around with orders on subsets and multiple-valued fixpoints, one may find a generalisation of Tarski's theorem to bifinite posets without $\perp$. The true generalisation, however, is as follows; we shall see why in §5.6.

Proposition Let X be a poset with directed sups, $r : X \to X$ a continuous coclosure with splitting $i : R \rightarrowtail X$, $p : X \to R$ and $f : X \to X$ a continuous function with $f \,;\, p = p$. Then there is a least continuous function $g : R \to X$ with $g \,;\, p = 1_R$ and $g \,;\, f = g$.

Proof Put $g = \bigvee^{\uparrow}(i \,;\, f^n)$. This is directed because $i \,;\, f \geq i \,;\, r \,;\, f = i \,;\, r = i$ so $i \,;\, f^{n+1} \geq i \,;\, f^n$. If h is another such function then $h \geq h \,;\, r = i$ so $h = h \,;\, f^n \geq i \,;\, f^n$ and $h \geq g$. □

$^+$**§2.4.9** We saw that the topology of bifinite posets is *almost* described by saying that they are coherent.

Proposition A distribitive lattice arises as the lattice of compact open sets of a (coherent) algebraic poset iff every element is expressable as a *finite* join of coprimes.

Proof Coprimes correspond to compact points of the poset and arbitrary elements of the lattice to compact open sets. In a coherent algebraic poset, compact opens are the upper-sets of finite sets of compact elements. □

Unfortunately the characterisation of the open set lattice of a bifinite posets is not so simple.

Theorem There is no first order theory whose models are precisely those distributive lattices whose spectrum is bifinite.

Proof Take the language $\mathcal{L}$ to be (include) 1, 0, $\curlywedge$, $\curlyvee$, $\leq$, $<$ (we can derive the last two as $x \leq y$ iff $x \curlywedge y = x$ and $x < y$ iff $x \leq y \ \wedge \ x \neq y$), together with constant symbols a_0, a_1,... and b_0, b_1,... Suppose $\mathcal{T}$ were the alleged first order theory. This would in particular include the usual equational presentation for distributive lattices. Recall that p is *prime* if $(\forall xy.x \curlywedge y \leq p)(x \leq p \ \vee \ y \leq p)$.

Now let Γ be the theory of distributive lattices together with the assertions that the a_i and b_i are distinct primes and

$$a_i \curlyvee b_i \;=\; a_{i+1} \curlywedge b_{i+1} \quad \text{(for each } i\text{)}$$
$$(\forall x)(\exists p, q \text{ prime})(x = p \curlywedge q)$$

Any model of Γ is the lattice of compact open sets of an algebraic poset in which any two compact elements have at most two mubs. However there is a pair of compact elements whose mub-closure is infinite, because the "radio mast" (example 2.4.4c) is embedded.

Hence $\Gamma \cup \mathcal{T}$ is inconsistent. So by the Compactness Theorem $\Gamma_n \cup \mathcal{T}$ has no model for some n, where Γ_n is the subset of Γ consisting of just those axioms involving suffices at most n.

But $\Gamma_n \cup \mathcal{T}$ has finite models: the corresponding posets are versions of the radio mast with only finitely many rungs.

Hence there is no such $\mathcal{T}$. □

$^+$§**2.4.10** Write $\mathsf{mubc}(X)$ for the collection of finite mub closed sets of compact elements of a bifinite poset X ordered by inclusion, and $\mathsf{Cocl}(X)$ for the poset of all continuous coclosures on X. Clearly we have naturally $\mathsf{Idl}(\mathsf{mubc}(X)) \subset \mathsf{Cocl}(X)$. From the characterisation of bifinite posets $\mathsf{mubc}(X)$ is a $\curlyvee$-semilattice and $\mathsf{Idl}(\mathsf{mubc}(X))$ an algebraic lattice whose top element is the identity.

Example Let X be a continuous but not algebraic lattice and $Y = \mathsf{Idl}X$. Then X is the image of a continuous coclosure on Y which is not a directed sup of finite ones.

Proof Recall $\downarrow\!\downarrow \dashv \bigvee^{\!\!\uparrow}$ made X a coclosure of Y (proposition 2.3.6). If this were in $\mathsf{Idl}(\mathsf{mubc}(Y))$ then X would itself be bifinite, and in particular algebraic. □

Question Is it true that if X is a retract of a bifinite poset that $\mathsf{Cocl}(X)$ is a continuous lattice? This was claimed in the original, but the "proof" depended on the assertion that codirected meets in a profinite poset distribute over directed joins.

Answer (Achim Jung, 4 December 1991) Michael Huth has the following result in his PhD thesis:

> Theorem 2.31 Let D be a bifinite dcpo. Then the following are equivalent:
>
> 1. $\mathsf{Cocl}(D)$ is continuous.
> 2. $\mathsf{Cocl}(D)$ is algebraic.
> 3. $\mathsf{Cocl}(D)$ is an algebraic lattice.
> 4. D is projection stable *i.e.* every coclosure has an algebraic image.

So every bifinite domain which is not projection stable is a counterexample to your proposition. Our student Magnus Rothe has generalised this to continuous domains:

> If D is a continuous domain and the space of all retractions is continuous then D is already algebraic.

I believe this also holds for the space of all coclosures. □

Cocl extends to a functor, indeed a monad, on **bcCont**, but not for general continuous domains.

$\mathsf{mubc}(X) \times \mathsf{mubc}(Y)$ generates $\mathsf{mubc}(X \times Y)$ or $\mathsf{mubc}(X \to Y)$ as a semilattice, and we may describe either of the latter as the "tensor product". The case for coproducts is simpler: $\mathsf{mubc}(X + Y) \cong \mathsf{mubc}(X) \times \mathsf{mubc}(Y)$. This is somewhat reminiscent of the behaviour of product topologies.

$^+$§**2.4.11** The interest in $\mathbf{BiPos}_\mathrm{f}$ lies in the fact that it is about as much of **AlgPos** as we can have whilst retaining cartesian closure. Plotkin [1976] conjectured that if X and X^X are countably based algebraic ipos then they are bifinite; we devote the remainder of this section to a revised version of Smyth's [1982] proof of this and slightly more. The techniques are *extremely* brutal, as are those we shall need to construct saturated domains. The original contribution in our presentation is essentially that we have twice as many lemmas as Smyth: we have shown separately that if each of the conditions fails then a generalised version of the corresponding counterexample is a retract, and that the function-space of the counterexample is not countably based algebraic. This result also differs from other "exponentiability" results such as proposition 2.3.8ζ in that it is more common to use 2 as the "test object"; however in this case 2^X is always an algebraic lattice whenever X is an algebraic poset.

For an ordinal α, write α^{op} and α^∇ for the algebraic ipos made of α "upside down" together with, in the second case, three extra elements $\{\bot, a, b\}$; *cf.* Example 2.4.4a.

Lemma

(a) Let $X = \alpha^{op}$ or α^∇. Then $X \to X$ is algebraic iff α is a successor ordinal.

(b) Let X be a poset with directed sups and some element x below which there is no minimal element. Then $\alpha^{op} \lhd X$ for some limit ordinal α.

(c) Let X be an ipo with $a, b \in X_{fp}$, $x \in X$ such that $a, b \le x$ and there is no mub of $\{a, b\}$ below x. Then $\alpha^\nabla \lhd X$ for some limit ordinal α.

Proof

[a,$\Rightarrow$] We show that if α is a limit then any $f \in [X \to X]$ (fixing a, b in the second case) is a proper directed sup, whence the identity is not approximable. For $\beta \in \alpha$ let $f_\beta(\gamma) = f(\gamma)$ if $\gamma \in \beta$ and $f(\gamma) + 1$ otherwise; then $f = \bigvee^{\!\!\uparrow}\{f_\beta : \beta \in \alpha\}$.

[a,$\Leftarrow$] If α is a successor then X, and hence X^X, is bifinite.

[b] Define $i : \alpha^{op} \hookrightarrow X$ for ordinals α. For $\alpha = 1$, let $i(0) = x$. For $\alpha = \beta + 1$, $i(\beta) \le x$ is not minimal, so let $i(\alpha) < i(\beta)$. For α a limit, carry on if we can, else stop. Now make α^{op} a retract. For $y \in X$ we cannot have $\forall \beta . y \le i(\beta)$ since otherwise the construction could continue, so let $\beta \in \alpha$ be $\in$-least ($\le$-greatest) with $y \not\le i(\beta)$ and put $p(y) = \beta$.

[c] Essentially similar to (b). □

Corollary If X and X^X are both algebraic ipos then X is mub-complete. □

$^+$§**2.4.12** Next we dispose of Example 2.4.4b; this time we make crucial use of countability.

Lemma

(a) Let X be a mub-complete algebraic poset with $\kappa \ge 2$ minimal elements. Then X^X has at least 2^κ minimal elements.

(b) Further let $Y \lhd X$. Then Y has at most κ minimal elements.

(c) Let X be a mub-complete algebraic ipo, and $a, b \in X_{fp}$ have $\kappa \ge 2$ mubs. Then there is a pair of compact elements of X^X with at least 2^κ mubs.

Proof

[a] Let M, N be the sets of minimal elements of X and X^X respectively, and let $0, 1 \in M$ be distinct with $0, 1 \leq \infty \in X$ if such a triple exists, otherwise any distinct $0, 1$. For $A \subset M$ define $A : X \to X$ by

$$A(x) \quad = \quad \begin{cases} 1 \text{ if } \exists a \in A. a \leq x \\ 0 \text{ if } \exists b \in M \backslash A. b \leq x \\ \infty \text{if both} \end{cases}$$

(we must have at least one by mub-completeness). Now suppose $n \leq A, B$ for $n \in N$. These are functions, so $n(m) \leq A(m), B(m)$ for all $m \in M$. But the restrictions of A and B to M are just the characteristic functions, so $A = B$. Hence 2^M is a quotient of a subset of N.

[b] Let y be minimal in Y and x minimal below iy in X. Then $px \leq p(iy) = y$ so $px = y$. Then the set of minimal elements of Y is a quotient of a subset of that for X.

[c] Same method as (a): consider the functions $[a \Rightarrow a]$ and $[b \Rightarrow b]$.. $\square$

Corollary

(a) Let Λ be an algebraic poset with $\Lambda^\Lambda \triangleleft \Lambda$. Then Λ is mub-complete and has $\bot$.

(b) Further suppose Λ is *not* coherent. Then $|\Lambda_{fp}| \geq \beth_\omega$, the first *strong limit cardinal*. This is the limit of $\beth_n$ where $\beth_0 = \omega$ and $\beth_{n+1} = 2^{\beth_n}$ ($\beth$ the Hebrew letter *beth*).

(c) Let X and X^X be countably based algebraic ipos. Then X is coherent. $\square$

Question

(a) Construct such an incoherent algebraic β-model.

(b) Is there a non-algebraic one without $\bot$?

$^+$**§2.4.13** It remains to bar Example 2.4.4c; for the purpose of this section, define a *radio mast* to be an infinite mub-complete algebraic poset with both $\bot$ and $\top$ and finite set S of points (not containing $\bot$ or $\top$) of which it is the mub-closure. The crucial property of such posets is that they have a *quasifinite* point, a non-finite point for which there is a finite set S of finite points such that every coclosure fixing S fixes the point.

Recall that for $a \in X_{fp}$, $[a \Rightarrow a]$ is the least continuous function fixing a and is compact in X^X; also these functions are incomparable for distinct $a \neq \bot$ (lemmas 2.3.3 and 2.6.6).

Lemma

(a) Let X be a coherent algebraic poset which is not bifinite; then X has a quasifinite point.

(b) Let X have a quasifinite point; then $Y \triangleleft X$ for some radio mast.

(c) Let X be a radio mast; then X^X is not algebraic.

Proof Let us first adopt some notation and terminology appropriate to all three parts. There is some set S of compact elements whose mub-closure is infinite; this has at least two but finitely many elements. Assign a *level* $h(x)$ to points in the mub-closure of S as follows:

$$h(x) \quad = \quad \begin{cases} 0 & \text{if } x = \bot \\ 1 & \text{if } x \in S \\ \leq n+1 & \text{if } x \text{ is a mub of two points at level } \leq n \\ \infty & \text{if } x \notin X_{fp} \\ \text{undefined} & \text{if } x \text{ is not in the mub-closure of } S \end{cases}$$

A quasifinite point is then one at infinite level.

[a] By a *chain* we mean a finite sequence $x_0 \leq x_1 \leq ... \leq x_n$ with $h(x_i) = i$; chains are partially ordered by amputation as usual and form a tree. There are at least two, but finitely many, chains with any given endpoint x_n, so this tree is infinite but finitely branching. Hence by König's lemma there is an infinite branch, being the initial segments of an infinite chain $m_0 \leq m_1 \leq ... \leq m = m_\infty = \bigvee^{\uparrow} m_i$ with $h(m_i) = i$. m is then a quasifinite point.

[b] Let $M = \downarrow m$ and Y be the set of points in the mub-closure of S which lie below m. This construction is unaltered if we now replace S by $S \cap Y$, so $S \subset Y \subset M$. Y is then a radio mast with generating set S. We shall define a retraction $f : X \triangleright Y$. For $x \in X_{fp}$ let $U(x) = Y \cap \downarrow x$, which increases monotonely with x; then put

$$f(x) \quad = \quad \begin{cases} y \text{ greatest with } y \in U(x) \text{ if } x \in M \\ m \text{otherwise} \end{cases}$$

Each case is monotone in x since $U(x)$ is, so we just have to check for $x \leq x'$ with $x \in M$ and $x' \notin M$ since M is down-closed. But then $f(x) = y \in U(x) \subset U(x') \leq m = f(x')$. f now extends uniquely to a continuous function on X which fixes Y.

[c] We shall show that the identity, 1_X, is neither compact nor approximable in X^X. By hypothesis X has a top element which not not compact in X. Hence the identity is not compact since $1_X \leq \mathsf{K}\top = \bigvee^{\uparrow}\{\mathsf{K}a : a \in X_{fp}\}$ but $1_X \not\leq \mathsf{K}a$ for any compact a. On the other hand, $[s \Rightarrow s]$ is a compact function below 1_X for $s \in S$, and there are at least two (but finitely many) such s. If X^X were algebraic, $X^X_{fp} \cap \downarrow 1_X$ would be directed, *i.e.* there would be a compact function g with $\forall s \in S.[s \Rightarrow s] \leq g \leq 1_X$. We show by induction on the level of x that $g(x) = x$, *i.e.* g is the identity, which is not compact. This is so for level 0 or 1 by hypothesis on g, so suppose x has level $n + 1$, being the mub of a and b at level at most n. Then $a = g(a) \leq g(x) \leq x$ (and similarly b), but x was a mub of a and b so $g(x) = x$. □

cf. Lamma 2.4.2.

Corollary If X and X^X are both countably based algebraic ipos then they are both bifinite. □

Proposition Let $\mathcal{C} \subset \mathbf{IPO} \cap \mathbf{AlgPos}_\omega$ be a full subcategory which is cartesian closed. Then $\mathcal{C} \subset \mathbf{BiPos}_\mathrm{f}$.

Proof Use lemma 2.1.9 to show that the product and exponential must be those from **IPO**. □

Note The largest cartesian closed category problem for algebraic and continuous domains with and without bottom has now been solved by Achim Jung. He discovered a new cartesian closed category, whose objects (called *L-domains*) are characterised by the property that $\downarrow x$ is a complete lattice for each $x \in X$, so Example 2.4.4b is one of them. Although he characterised two maximal cartesian closed subcategories of continuous ipos (one of them consisting of continuous L-domains), it remains (January 1992) an open question whether the other is strictly larger than the "continuous domains" defined in the next section — almost certainly it is.

2.5 The Hierarchy of Categories of Domains

+§**2.5.1** In this section we explore some further important categories of domains and perform constructions in them. We see a few glimpses of a very rich structure which has yet to be uncovered. The methods I have in mind for investigating this hierarchy of categories of domains further again illustrate the deliberate confusion of object and meta-levels which is a necessary feature of any discussion of a "type of types". For whereas, on the one hand, we shall see the suggestion of a correspondence between categories of domains and fragments of (coherent) logic (*via* the classifying topos, which we shall discuss in §3.3), on the other hand it seems that coherent logical properties of the domains themselves are the appropriate tool for classifying the categories.

$^+$§**2.5.2** My hope in this work was to define a domain to be a certain kind of continuous category (§2.3.6), and consider arbitrary categories of them (and continuous functors) which are closed under products, retracts, exponentials and (countable) bilimits. Unfortunately the additional theory needed for categories rather than posets far exceeds what was feasible, and even the theory for bifinite posets proved more difficult than expected. We have already seen the majority of the Domain Theory to be considered in this work (we shortly turn our attention to Indexed Category Theory), but for the rest of it we concentrate mainly on $\mathbf{bcCont}_\omega$.

Question Is the class of retracts of bifinite posets closed under bilimits?

Answer Yes. For a proof see Theorem 4.6 of [Jung 1987].

Because of this embarrassing gap, we define **ContDom**, the category of *continuous domains*, to be the full subcategory of **IPO** generated by $\mathbf{BiPos}_\mathrm{f}$ together with retracts and bilimits. $\mathbf{ContDom}_\omega$ consists of the countably based such.

Lemma **ContDom** and $\mathbf{ContDom}_\omega$ are cartesian closed.

Proof Products and exponentials essentially commute with retracts (proposition 1.3.5&6) and bilimits (proposition 2.2.5). □

Definition Let $\mathcal{C}$ be a full subcategory of **ContDom** closed under products, exponentials and countable bilimits. Then $\mathcal{C}$ is a

(i) *small category of algebraic domains* if $\mathcal{C} \subset \mathsf{Bi}_\omega\,\mathbf{Pos}_f$ and is closed under retracts *in so far as they exist there*,

(ii) *small category of continuous domains* if $\mathcal{C} \subset \mathbf{ContDom}_\omega$ and is closed under retracts,

(iii) *large category of algebraic domains* if $\mathcal{C} \subset \mathbf{BiPos}_\mathrm{f}$ and is closed under arbitrary small bilimits and retracts there,

(iv) *large category of continuous domains* if $\mathcal{C} \subset$ **ContDom** and is closed under bilimits and retracts.

We shall describe these variant definitions as *flavours*.

By the *weight* of a domain we mean the least cardinality of a *base*, *i.e.* set which generates the domain by directed sup. For an algebraic domain X this of course means the cardinality of X_{fp}. The *weight* of a *category* of domains is meant the supremum of the weights of its objects.

§**2.5.3** By far the most widely used category of domains is $\mathbf{bcAlg}_\omega$, whose objects are countably based algebraic posets in which any bounded subset has a least upper bound. One may feel confident that this is an appropriate tool for the study of the semantics of deterministic recursive programs. Following our preference for closure under retracts, the corresponding $\mathbf{bcCont}_\omega$ is of more interest to us; similarly we have the other flavours **bcAlg** and **bcCont**.

From the point of view of geometric logic, spaces in **bcAlg** classify models of *Horn theories*. These have as axioms sequents (*i.e.* statements of the form $\phi \vdash \psi$, where the "implication" is of an *external* nature and cannot be embedded) whose antecedent and consequent are built up from atomic formulae using only *true*, *false*, *and* and *exists*. (Full *coherent* logic allows *finite* disjunction and *geometric* logic infinite disjunction: see §3.3.11.)

Since the conjunction of *false* ($\bot$) with anything else is still $\bot$ and any sequent with $\bot$ as antecedent is redundant, we may divide the axiomatisation into two parts, the first being *essentially algebraic*, *i.e.* in which the only connective is finite conjunction, and the second consisting of negations of algebraic formulae. Let us call these the *positive* and *negative* parts; more specifically we have a pair of sets **P** and **N** of the form

$$\mathbf{P} = \{(\Gamma_i, \phi_i) : i \in I\} \qquad \mathbf{N} = \{\Delta_j : j \in J\}$$

where ϕ_i are atomic formulae and Γ_i, Δ_j are finite sets of atomic formulae. The axiomatisation is *inconsistent* if, for some $j \in J$, we can deduce each $\delta \in \Delta_j$ from $\mathbf{P}$. Alternatively, we may deduce the further sequent $\Gamma \vdash \Delta$ from $(\mathbf{P}, \mathbf{N})$ if we reach inconsistency by adding $\{(\emptyset, \gamma) : \gamma \in \Gamma\}$ to $\mathbf{P}$ and $\{\Delta\}$ or nothing to $\mathbf{N}$ if Δ is respectively algebraic or false.

This fragment of logic has been implemented computationally as the programming language PROLOG [Clocksin & Mellish 1981]. In this case $\mathbf{P}$ is usually larger than $\mathbf{N}$ and is called the *database*, whilst $\mathbf{N}$ and possible deductions from $(\mathbf{P}, \mathbf{N})$ are called *queries*. The driving force of a PROLOG interpreter is a *unification algorithm*, which attempts to find simultaneous substitutions into pairs of terms to make them coincide (syntactically); this enables us to search for applications of axioms and hence develop the proof tree. By this means we determine whether or not a given atomic formula is deduceable from the database.

Scott [1982] has also investigated this fragment of logic and shows by means of *information systems* that it does indeed correspond to the category $\mathbf{bcAlg}_\omega$.

§**2.5.4** The least nontrivial fragment of coherent logic is *essentially algebraic logic*, in which $\perp$ is excluded. Clearly the notion of inconsistency disappears, though of course we may have only the singleton model. The corresponding category of domains is **AlgLat** (or **ContLat** or $\mathbf{AlgLat}_\omega$ or $\mathbf{ContLat}_\omega$ according to taste); the categorical analogue of this is described by *Gabriel-Ulmer duality* [1971].

Next after this is the logic of *partial functions*. The domains arising as spaces of models here have the property that for any subset, if every pair from that subset is bounded (consistent) then the whole subset has a least upper bound. The word *coherent* has been used in the computer science literature to describe these, but again we prefer to reserve this for its established use in logic, category theory and topology. Because of the connection with partial functions we shall call them *partial lattices* and write **PAlg**, **PCont**, $\mathbf{PAlg}_\omega$ and $\mathbf{PCont}_\omega$ for the flavours. Write $x\#y$ for $\neg\exists z.x \leq z \geq y$.

Proposition (Plotkin 1978) $\mathcal{T}^\omega$ is a saturated object for $\mathbf{PCont}_\omega$, where $\mathcal{T}$ is the three-element poset looking like a V.

Proof We have to show that if $X \in \mathbf{PCont}_\omega$ then $X \triangleleft \mathcal{T}^\omega$. We represent elements of $\mathcal{T}^\omega$ by a pair of disjoint subsets of $\mathbb{N}$. Let e_n enumerate a basis for X, so that $\forall x \in X.x = \bigvee^{\uparrow}\{e_n : e_n \ll x\}$. Let $\phi : X \to \mathcal{T}^\omega$ by

$$x \quad \mapsto \quad \langle \{n : e_n \ll x\}, \{m : e_m \# x\}\rangle$$

and $\psi : \mathcal{T}^\omega \to X$ by

$$\langle u, v\rangle \quad \mapsto \quad \bigvee\nolimits^{\uparrow}\{e_n : n \in u \wedge \forall m \leq n.e_m \# e_n \Rightarrow m \in v\}$$

□

Since in particular its own function space is a partial lattice and therefore a retract, this has a β-model structure; in fact this is a corollary of our theorem 2.6.11c. Barendregt and Longo [1980] have shown that the theory of equality in this model is the same as that in the Böhm tree model.

It is a further consequence of Plotkin's result that the posets of partial maps between objects of suitable categories are exactly partial lattices. From proposition 2.6.4b it will follow that countability is not necessary in the above result, answering the question in Plotkin's concluding remarks.

+§**2.5.5** We shall now give a hint of the application of coherent logic to classifying the hierarchy of domains. Let ϕ be a coherent formula in the language of posets with n free variables $\vec{x}$, so it is a finite disjunction of formulae of the form $\exists\vec{y}.\alpha(\vec{x}, \vec{y})$ where $\vec{y}$ has length m and α is a finite conjunction of instances of the order relation between variables. Then we write $(A, \vec{a}) \models \exists\vec{y}.\alpha(\vec{x}, \vec{y})$, where A is a poset and $\vec{a} \in A^n$, if $\alpha(\vec{a}, \vec{b})$ holds for some $\vec{b} \in A^m$, and likewise for disjunctions.

Lemma Let $f : A \to B$ be a monotone map and $(A, \vec{a}) \models \phi$ for some coherent formula ϕ with n free variables and some $\vec{a} \in A^n$. Then $(B, f(\vec{a})) \models \phi$. □

Proposition Let $A \lhd \mathsf{bilim}\, A_i$ with A_i finite, $p_i : A \to A_i$ the obvious maps, ϕ a coherent formula with n free variables and $\vec{a} \in A^n$. Then $(A, \vec{a}) \models \phi$ iff for each i, $(A_i, p_i(\vec{a})) \models \phi$.

Proof Put $\vec{a}_i = p_i(\vec{a})$. [$\Rightarrow$] follows easily from the lemma, as does [$\Leftarrow$] for retracts; we split [$\Leftarrow$] for bilimits into the cases involving conjunctions of atomics (follows by definition), the existential quantifier and finite disjunction.

[$\exists$] Let $\phi = \exists \vec{y}.\alpha(\vec{x}, \vec{y})$ as above, and suppose for each i, $(A, \vec{a}_i) \models \phi$. Then $W_i = \{\vec{b} \in A_i^m : \alpha(\vec{a}_i, \vec{b})\}$ is nonempty and finite. By the lemma, the projection $A_i \to A_j$ restricts to $W_i \to W_j$. This is a cofiltered diagram of nonempty finite sets, whose limit is nonempty by lemma 1.2.12b; let $\vec{b}$ be in it. Since we have projections, $\vec{b} = \bigvee^{\uparrow} \vec{b}_i$; then $(A, \vec{a}, \vec{b}) \models \alpha$, so $(A, \vec{a}) \models \phi$.

[$\vee$] Now let $\phi = \bigvee_{j \in J} \phi_j$ be a finite disjunction, with each ϕ_j of the form in ($\exists$), and for each i, $(A_i, \vec{a}_i) \models \phi$. Let $I_j = \{i : (A, \vec{a}_i) \models \phi_j\}$; these sets are down-closed in the diagram by the lemma and cover I. But I is filtered and J finite, so $I_j = I$ for some j. Hence $(A, \vec{a}_i) \models \phi_j$ by ($\exists$). □

Question Does this result extend to geometric formulae?

$^+$§**2.5.6** There are in fact many more categories of domains lying between $\mathbf{ContLat}_\omega$ and $\mathbf{bcCont}_\omega$. Write H_n for the "topless $n+1$-diamond", *i.e.* the poset of *proper* subsets of an $n+1$-set. Also write P_r^s for the property (of a poset) that for any s-set, if any r-subset of it is bounded then the whole s-set has a least upper bound; observe that this is a coherent sequent.

Lemma

(a) if $s' \le s$, $X \models P_r^s$ and $r \le r'$ then $X \models P_{r'}^{s'}$.

(b) H_n satisfies P_r^s for all n, r, s unless $r < n+1 \le s$.

(c) if $X \models P_r^s$ and $Y \lhd X^U$ then $Y \models P_r^s$. □

Proposition Given $\Gamma \subset \{P_r^s : r, s \in \mathbb{N}\}$, let $\mathcal{C}$ be the full subcategory of **IPO** whose objects satisfy Γ and are retracts of countably based bifinite posets. Then $\mathcal{C}$ is a small category of continuous domains, and there are infinitely many such $\mathcal{C}$.

Proof By part (c) of the Lemma, and proposition 2.5.5, $\mathcal{C}$ is closed under exponentials, retracts (hence products by corollary 1.3.5) and countable bilimits. □

Gordon Plotkin claims that the H_n classify categories of boundedly complete domains. The crucial point is that any finite boundedly complete poset is a retract of a power of H_n, where $n+1$ is the size of the largest minimally inconsistent subset. The proof of this is presumably a generalisation of that of proposition 2.5.4, but unfortunately I was unable to locate it in his notes.

$^+$§**2.5.7** As a further corollary of this application of coherent logic, we have a result about the subspace topology on $\mathsf{Max}\, X$, the set of maximal elements of a bifinite poset X.

The interest in $\mathsf{Max}\, X$ lies in the idea [which is plausible at the first order but breaks down for higher order functions, where the logical complexity of the totality condition increases] that it represents the "total" or "terminating" elements of the domain, the others being "partial" or "nonterminating".

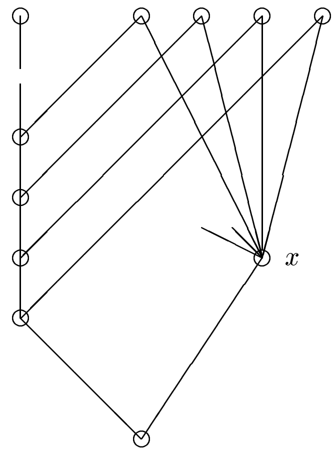


Figure 2.5.7: $\mathsf{Max} \cap \uparrow x$ is not closed in Max

Lemma Let $X \in \mathbf{ContDom}$ and $x \in X$. Then $\downarrow(\uparrow x)$ and $\uparrow x \cap \mathsf{Max}\, X$ are closed in X and $\mathsf{Max}\, X$ respectively.

Proof Following §2.5.2, we have to prove this by induction on the construction of X. It is clearly true for finite posets, so let X be a retract of a bilimit of domains X_i in which the result holds and $p_i : X \to X_i$ the appropriate maps. Now "$y \in \downarrow\uparrow x$" is a coherent formula in x, y, so by proposition 2.5.5, $\downarrow(\uparrow x) = \bigcap p_i^{-1}(\downarrow(\uparrow p_i x))$, which is an intersection of inverse images of closed sets and hence closed. The result for $\mathsf{Max}\, X$ follows from the definition of the subspace topology.

Proposition Let X be bifinite. Then $\mathsf{Max}\, X$ is zero-dimensional.

Proof If $x \in X_{fp}$ then $\uparrow x \cap \mathsf{Max}\, X$ is clopen, and these sets form a base for the topology. □

Example

(a) In an infinite flat domain, Max is not compact.

(b) In this algebraic poset there is a point x with $\uparrow x \cap \mathsf{Max}$ not closed.

$^+$§**2.5.8** Lattices, partial lattices, boundedly complete posets and the other intermediate categories of domains correspond to subfragments of propositional Horn logic. In the other direction we may investigate what happens when we add disjunction to our fragment — necessarily leading to nondeterministic programs — or move to full model theory in stead of just propositions. The former leads to larger categories of posets (a certain fragment of coherent logic gives all bifinite posets, but as we saw in 2.4.9 it is not full coherent logic and the question as to what fragment it is is not an easy one) and the latter to categories or toposes.

We define a *bifinite category* to be one which is a bilimit of finite categories. Much of the theory of bifinite posets carries over immediately, though there are pitfalls.

Fact The category of algebraic field extensions of $\mathbb{Q}$ is bifinite (but is not a poset). □

The mind boggles at the thought of Galois models of the λ-calculus! However there is nevertheless a moral to be drawn from this example. The interest of Algebraic Number Theory lies in the objects of the above category, but it is sometimes convenient to work inside the algebraic closure, $\mathbb{A}$, of $\mathbb{Q}$ (or even in the complex numbers, $\mathbb{C}$). This serves the same function as a saturated domain in our subject, and indeed the two constructions are much the same. Our interest lies in types, *i.e.*

objects of the category, but we shall find it convenient to work in a saturated domain, Λ (since it is necessarily a β-model). The choice of a *particular* algebraic closure of $\mathbb{Q}$ is irrelevant to algebraic number theory, and we likewise shall regard the *particular* choice of Λ to be unimportant. (There is actually a distinction in that any two algebraic closures of $\mathbb{Q}$ are isomorphic.)

§**2.5.9** For the remainder of this section we concern ourselves only with **bcCont**.

Proposition The forgetful functor $\mathbf{bcCont}^{hm} \to \mathbf{bcCont}$ has a left adjoint, $\mathcal{F}$. Moreover the unit $X \to \mathcal{F}X$ is an embedding.

Proof For $x \in X \in \mathbf{bcCont}$, let $\mathcal{F}X$ be the poset of *proper* filters of open subsets of X and $\eta_X x$ the collection of open neighbourhoods of x. [Note: $\mathcal{F}X$ is then a set of sets of sets; a filter is proper iff it does not contain the empty set.]

Given $f : X \to Y$ in **bcCont** we have a homomorphism $\mathcal{F}X \to Y$ by

$$\begin{array}{llll} f^* : Y \to \mathcal{F}X & \text{by} & y & \mapsto \{U : (\exists V : y \in V)(f^{-1}V \subset U)\} \\ f_* : \mathcal{F}X \to Y & \text{by} & \Phi & \mapsto \bigvee^{\uparrow}\{\bigwedge\{fx : x \in U\} : U \in \Phi\} \end{array}$$

Then $f^* \dashv f_*$ since $f^*y \subset \Phi$ and $y \leq f_*\Phi$ are each equivalent to $(\forall V : y \in V)(f^{-1}V \subset \Phi)$. Also f_* is continuous, $f = \eta_X \,;\, f_*$ and $\eta_X = 1^*$. □

Question Does this hold for other categories of (boundedly complete) continuous domains?

§**2.5.10** There is a characterisation of boundedly complete continuous posets similar to that for continuous lattices; indeed much of the theory of one category is obtained directly from the other by minor surgery.

Recall from §2.3.1 that a space is *injective* if any continuous map to it from a *subspace* of another space can be extended to the whole of that space. More generally a T_0 space is *densely injective* if the same holds for *dense* subspaces.

Proposition The following are equivalent for a T_0 space X equipped with its specialisation order:

(α) X is a boundedly complete continuous poset with its Scott topology,

(β) X is a closed subspace of a continuous lattice,

(γ) $X \cup \{\top\}$, where $\forall x.x \leq \top$, is a continuous lattice

(δ) X is densely injective

(ϵ) X is an "algebra" for arbitrary *inhabited* inf distributing over directed sup.

Proof Only dense injectivity is not *completely* obvious.

[$\gamma \Rightarrow \delta$] Let $A \subset B$ be dense. We may extend any $A \to X$ to $B \to X \cup \{\top\}$. X is closed in the codomain, so its inverse image, which contains A, is in the domain. But A is dense, so $B \to X$.

[$\delta \Rightarrow \gamma$] Let $A \subset C$, with B the closure of A. Extend $A \to X$ to $B \to X$ by dense injectivity, and to $C \to X \cup \{\top\}$ by $C \backslash B \to \{\top\}$. □

Considerations from Universal Algebra strongly suggest that we admit the empty poset to **bcCont**, but we resist them.

Warning $\mathbf{bcCont}^{hm}$ is *not* algebraic over **Set**; indeed it is not even complete.

[+]**§2.5.11** Here is something we promised in Example 1.5.7 (this I now understand to be Martin Hyland's original construction from 1973).

Lemma Let $X \in \mathbf{ContDom}$. Then the diagonal map $X \to X \times X$ is an embedding iff X is boundedly complete.

Proof The right adjoint to the diagonal is the binary inf operation. Every continuous domain has directed inf, so this exists iff the domain is boundedly complete. Binary inf is continuous since the domain is continuous. □

Proposition In **bcCont** there are nontrivial objects Λ with $\Lambda \cong \Lambda \times \Lambda \cong \Lambda^\Lambda$; indeed any object is a retract of such.

Proof Let $X_0 \in \mathbf{bcCont}$. Put $X_{2n+1} = X_{2n} \times X_{2n}$, embedding X_{2n} by the diagonal map. Put $X_{2n+2} = {X_{2n+1}}^{X_{2n+1}}$, embedding X_{2n+1} by dropping a variable. Finally put $\Lambda = \mathsf{bilim}\, X_i$. Then $\Lambda \cong \mathsf{bilim}\, X_{2n+1} \cong \mathsf{bilim}[X_{2n} \times X_{2n}] \cong [\mathsf{bilim}\, X_{2n}] \times [\mathsf{bilim}\, X_{2n}] \cong \Lambda \times \Lambda$ by continuity of $\times$. Similarly $\Lambda \cong \Lambda^\Lambda$. Also $X_0 \lhd \Lambda$. □

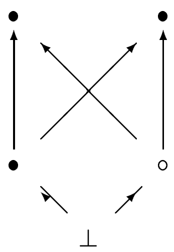
The Y combinator gives least fixed points for the same reason as it did in proposition 2.2.8b.

We cannot in general expect to be able to solve double fixpoint equations (§1.5.11), but the relationship between $\times$ and $\to$ enables us to find $X \lhd \Lambda \cong \Lambda \times \Lambda \cong \Lambda^\Lambda$. For let $U = X^{\mathbb{N}}$ and solve $\Lambda \cong U^\Lambda$. Then $U \cong U \times U$, $\Lambda \cong U^\Lambda \cong (U \times U)^\Lambda \cong U^\Lambda \times U^\Lambda \cong \Lambda \times \Lambda$ and $\Lambda^\Lambda \cong (U^\Lambda)^\Lambda \cong U^{(\Lambda \times \Lambda)} \cong U^\Lambda \cong \Lambda$.

[+]**§2.5.12** A large part of the difficulty in extending results about algebraic domains to the continuous case is that no intrinsic characterisation has yet been found of the retracts of bifinite posets. The problem is really that we have far less grasp on a general retract than on closure and coclosure operators; for lattices we overcame this by factoring a retract as a composite of a closure and a coclosure (this is possible because the intermediate retract is the join of the given one with the identity in a certain lattice — perhaps there's some kind of "zig-zag" lemma in general).

Unfortunately Idl, which we used in lemma 2.3.6, preserves mub-completeness but not bifiniteness.

Example Let X be the countable power of the poset



Then the pair of sequences $\{a^\infty, b^\infty\}$ in $\mathsf{Idl}X$ has infinitely (indeed uncountably) many mubs, namely those sequences each of whose terms is either c or d. □

However if $X \in \mathbf{bcCont}$ then $\mathsf{Idl}X \in \mathbf{bcAlg}$.

§2.5.13 Somewhat after the submission of this dissertation, Gordon Plotkin told me of his characterisation of continuous domains. We say a domain X is *gradable* if there is an increasing sequence of endomorphisms $\delta_i : X \to X$ (not necessarily idempotent) with $\bigvee^{\uparrow} \delta_i = 1_X$ and $\delta_i(X)$ finite.

It is easy to see that retracts, products and exponentials of gradable domains are gradable, and that countably based bifinite posets are gradable.

$\delta_i(x) \ll x$, and $x \in X_{fp} \iff \exists i. \delta_i(x) = x$. It is not, however, necessarily the case that $\delta_i \ll 1_X$.

Proposition (a) Any gradable domain is a retract of a countably-based bifinite poset.

Proof Let δ_i grade X. Put

$$Y_i \quad = \quad \{\langle u_0, u_1, ..., u_i\rangle : u_j \in \bigcup_{k \le j} \delta_k(X), u_k \le u_j : (k \le j \le i)\}$$

which is finite; also put Y for the similar set of infinite sequences. These form a bilimiting diagram in which the projections curtail the sequences and the embeddings repeat the last term, so Y is bifinite. We make $X \lhd Y$ by $x \mapsto \langle \delta_i(x)\rangle$ and $\langle u_i\rangle \mapsto \bigvee^{\uparrow} u_i$. □

Corollary A domain is gradable iff it is a retract of a countably based bifinite poset. □

By modifying the definition to say that the deflationary endomorphisms with finite image form a directed set with sup the identity, we should be able to extend this beyond the countably-based case.

Proposition (b) A bilimit of gradable domains is gradable.

Proof W.l.o.g. the bilimit is of embeddings. I can only see how to prove the result using one of the variant definitions, so that the $i+1$st function dominates the preceding ones. □

Theorem The retracts of (countably based) bifinite posets form a category of domains, and in particular have bilimits. □

2.6 Saturated Domains

§**2.6.1** We devote the final section of this chapter on domain theory to the construction of *saturated domains*, *i.e.* of which any other occurs as a retract (the word *universal* appears in the literature, but I take the view that this should be reserved for its categorical sense involving uniqueness). This is done for $\mathsf{Bi}_\omega\,\mathbf{Pos}_f$, though the technique is more widely applicable.

Finding saturated domains is really part of the more comprehensive question of showing that small categories of continuous domains and categories of retracts of continuous models of the λ-calculus are one and the same thing. The proof of this in the generality I would have liked has several gaps, but it is shown here for subcategories of $\mathbf{bcCont}_\omega$.

The point of this, as remarked in §2.5.8 by analogy with algebraically closed fields, is that whilst it is the *category* of domains which is our primary interest it is convenient to assume that it is of the form $\mathbf{Retr}(\Lambda)$. In chapter V we shall make substantial use of this assumption to motivate our definition and subsequent construction of a type of types.

In contradistinction to the case of algebraically closed fields, the technique is a "bootstrapping" one: we make use of any β-model we are given to construct another with slightly better properties. As has been repeatedly emphasised, the primary interest is *not* in the structure of any *particular* model. Moreover since each application of this technique suggests a new desirable property of models of the λ-calculus we make no attempt to formulate the optimal definition.

$^+$§**2.6.2** The early constructions of models of the λ-calculus overcame the obvious cardinality problem with discrete sets because the step functions of §2.3.10 provide a handle on the size of the function space. The idea of the construction of saturated domains is much the same: we use the finite posets as a (countable) "basis" for the category.

Lemma Let Y be a finite poset and $X = \mathsf{bilim}\, X_i$ in **IPO**. Then Y occurs as a retract of X iff it occurs as a retract of some X_i.

Proof [$\Leftarrow$] is obvious. [$\Rightarrow$] We use the characterisation of finite retracts from proposition 2.1.11b. For $y \in Y$ let U_y and A_y be the inverse images of $\uparrow y$ and $\downarrow y$, so $y \in U_y \cap A_y$. For $i \in I$ let $p_i : X \to X_i \subset X$ be the corresponding projection, so $\bigvee^{\uparrow}\{p_i : i \in I\} = 1_X$ (*cf.* proposition 2.4.3β).

Then for $y \in Y$, $y = \bigvee^{\uparrow}\{p_i y : i \in I\}$, so for some $i = i_y$, $p_i y \in U_y$. There are only finitely many y, so choose $i \geq i_y$. Then $Z = \{p_i y : y \in Y\}$ is the image of Y in X_i and is a choice satisfying the conditions of proposition 2.1.11b. Hence it is the image of a retract on X with the same partition (and $Y \cong Z$). This retraction restricts to X_i, although the composite $Y \cong Z \lhd X_i \lhd X$ need not be the given retraction. □

Proposition Let X, Y be finite posets. Then Y is an object of any category of domains which also contains X iff $Y \lhd X_m$ for some m.

Proof

[$\Leftarrow$] Any category of domains is by definition closed under retracts and exponentials.

[$\Rightarrow$] Let Σ be the class of (finite) posets which occur as retracts of iterated function spaces of X. By the lemma, if we extend Σ to a category $\mathcal{C}$ by alternately adjoining bilimits and retracts we add no new *finite* domains. Any such extension remains closed under exponentials (lemma 2.5.2). The closure under such operations is a category of domains, indeed the smallest which also contains X, so it contains Y by hypothesis. Hence $Y \in \Sigma$. □

$^+$§**2.6.3** We would like to show that any category of domains is determined by its finite objects. There is no problem with algebraic domains, but we need to show for continuous domains that we can modify any expression of the form $X \lhd \mathsf{bilim}\, Y_i$ to replace the Y_i by Z_i which are necessarily in any category of domains which contains X.

The best I can do here is by restriction to boundedly complete continuous posets. By a *subsemilattice* is meant a subset closed under such binary joins as exist.

Lemma

(a) Let $X \in \mathbf{bcCont}$ and $S \subset X$ be a finite subsemilattice. Then $S \lhd X$.

(b) Let $X \in \mathbf{bcCont}_\omega$. Then $X \lhd \mathsf{bilim}\, Y_i$ for some countable diagram of finite posets Y_i which are retracts of X.

(c) Let $X \in \mathbf{bcCont}$ and U be any finite poset. Then $X^U \lhd X^n$ for some n.

Proof

[a] For $s \in S$ let $U_s = \{x : (\forall t \in S : x \leq t)(s \leq t)\}$. Then $s \in U_s$, $U_\perp = X$, and $s \leq t$ iff $U_s \supset U_t$. Also $x \in U_s \cap U_t$ iff $(\forall u \in S.x \leq u)(s,t \leq u)$; hence $U_s \cap U_t = U_{s \curlyvee t}$. Now let $p : X \twoheadrightarrow S$ by $p(x) = \bigcap\{s : x \in U_s\}$.

[b] Let B be a countable base for X, which we may as well assume to be a subsemilattice. Let S be any finite subsemilattice of B; then S is a retract of X by the Lemma. Then $Y = \mathsf{Idl} B \cong \mathsf{bilim}\, S$ so it suffices to show $X \lhd Y$. But B was a base, so any $x \in X$ is a directed sup from B, hence an element of Y, and the inclusion $B \subset X$ extends to a continuous map $Y \to X$. Thus X is in fact a coclosure of Y. Of course we have the same result with "countable" deleted.

[c] Given $f : |U| \to X$ (where $|U|$ means the underlying *discrete* set of U), let $g(u) = \bigwedge\{f(q) : u \leq q\}$. Then $X^U \lhd X^{|U|}$ by a coclosure. □

$^+$§**2.6.4** We now have two forms of the desired result, a weak one for general algebraic (*i.e.* bifinite) domains and a strong one for boundedly-complete continuous domains.

Proposition (a) There is a bijective correspondence between

(α) large or small categories $\mathcal{C}$ of algebraic domains,

(β) classes Σ of finite posets closed under exponentials and retracts.

Proof Put $\Sigma = \mathcal{C} \cap \mathbf{Pos}_f$ and $\mathcal{C} = \mathsf{Bi}_\omega\,\Sigma$. Then $\Sigma \mapsto \mathcal{C} \mapsto \Sigma$ is trivially the identity and $\mathcal{C} \mapsto \Sigma \mapsto \mathcal{C}$ is because any bifinite poset can be expressed as a bilimit of retracts. Σ is of precisely this form by proposition 2.6.2. $\square$

Proposition (b) There is a bijective correspondence between

(α) large or small categories of boundedly-complete continuous domains,

(β) classes of finite boundedly-complete posets closed under products and retracts.

Proof Further use lemma 2.6.3b for equivalence and lemma 2.6.3c for the form of Σ. $\square$

$^+$§**2.6.5** Let $\mathcal{H}$ be the poset of categories of algebraic domains under inclusion. We call this the *hierarchy*.

Theorem

(a) $\mathcal{H}$ is an infinite countably based algebraic lattice.

(b) The basis (subposet of compact elements) of the hierarchy below **bcAlg** is recursively decidable.

Proof

[a] This is an immediate corollary of proposition 2.6.4a since exponentiation and retracts are finitary operations; infinity follows from proposition 2.5.6.

[b] We have only to test, for finite boundedly-complete posets X, Y, whether $Y \triangleleft X^n$ for some n. However it is easy to show that w.l.o.g. $n = |X^Y|$. *cf.* §2.5.6. $\square$

Question Is it distributive, and if so, what does its spectrum mean?

[My reason for suspecting this is the application of coherent logic to the classification (Question 2.5.5): points in the hierarchy appear to correspond to Grothendieck topologies on $\mathbf{Pos}_f$ and hence to elements of its locale of nuclei.]

It may be useful to apply (abstract) homology theory to the classification of domains, for instance making the simplicial sets at dimension n the bounded n-tuples of the domain (this idea as it stands unfortunately gives a zero result), although as we noted in §2.5.6, Gordon Plotkin claims that boundedly complete domains have essentially one invariant of products, retracts and bilimits, namely "dimension". A categorical union of homology and coherent logic, incorporating a generalisation of the notion of dimension, would be needed for general bifinite posets.

Remark See (per Archim Jung)

- Baclawski, Bjoerner: Fixed Points in Partially Ordered Sets, *Adv. Math.*__31__ (1979) 263–287.

- Walker: Isotone Relations and the Fixed Point Property for Posets. *Discrete Math.*__48__ (1984) 275–288.

$^{+}$§**2.6.6** At this point we temporarily shift emphasis from categories of domains back to models of the lambda calculus and their retracts. First we get a grasp of the size of such a model. Recall from §§2.3.1-2 that the Scott topology on a domain X is 2^X.

Lemma Let X be an algebraic ipo and $a, b \in X_{fp}$. If $[a \Rightarrow a] \leq [b \Rightarrow b]$ then either $a = \bot$ or $a = b$.

Proof First note $b \not\leq a$ iff $\exists x. a \leq x \not\geq b$; under our hypothesis if this holds then $a \leq \bot$, otherwise $b \leq a$. In either case $a \leq y \geq b$ for some y, whence the hypothesis gives $a \leq b$. $\square$

Proposition Let X be an algebraic ipo with weight $\kappa \geq \omega$. Then $2^\kappa \lhd X_2$.

Proof The topology of X_1 is a retract of X_2. By the lemma $(X_1)_{fp}$ has an antichain (discrete subspace) of cardinality κ, so the topology of X_1 has 2^κ (the topology of a discrete space of size κ) as retract. $\square$

Corollary Let Λ be a β-model in **AlgPos** of weight κ. Then the *flat lattice*, $\kappa_\bot^\top$, of size κ (obtained by adding top and bottom to a discrete set) is an object of $\mathbf{Retr}(\Lambda)$. So is the (ordinal) *chain* of length $\kappa + 1$.

Proof These are retracts of 2^κ. $\square$

$^{+}$§**2.6.7** We can use this lattice to index products if we have $\top$ in the model; otherwise we want the *flat domain*, $\kappa_\bot$, obtained by adding just $\bot$ to a discrete set. Unfortunately I cannot see how to get beyond ω.

Lemma Let $X \in$ **ContPos** have bottom but not top. Then

(a) $\mathcal{T} \lhd X$, where $\mathcal{T} = \{0, 1, \bot\}$ is the poset introduced in §2.5.4

(b) If $U, V \lhd X$ then $U +_\bot V$ (obtained by adding a new $\bot$ to the disjoint union of U and V) is a retract of $X \times X$.

Proof

[a] Let $0, 1 \in X$ be unbounded; choose $0' \ll 0$, $1' \ll 1$ with $0' \not\leq 1$ and $1' \not\leq 0$. Then take $\twoheaduparrow 0'$ to 0, $\twoheaduparrow 1'$ to 1 and the rest to $\bot$.

[b] (*cf.* §1.4.6) Write $T, U, V : X \to X$ for the corresponding idempotents and define an endofunction of $X \times X$ by

$$(x_1, x_2) \quad \mapsto \quad \begin{cases} Ux_2 \text{if } Tx_1 = 0 \\ Vx_2 \text{if } Tx_1 = 1 \\ \bot \quad \text{if } Tx_1 = \bot \end{cases}$$

$\square$

Proposition Let Λ be a β-model in **ContPos** without top. Then the *flat domain*, $\omega_\bot$, of size ω (add bottom to a discrete set) is an object of $\mathbf{Retr}(\Lambda)$.

Proof Let $F = \lambda X. X +_\bot (\mathsf{K}\bot)$. Then $F^n(\mathsf{K}\bot)$ has the shape indicated by the figure.
$\omega_\bot \lhd \bigvee^{\uparrow}\{F^n(\mathsf{K}\bot)\}$ by sending all non-open points to $\bot$. $\square$

$^{+}$§**2.6.8** We can use the previous two results to construct countable products in $\mathbf{Retr}(\Lambda)$.

Proposition Let Λ be a β-model in $\mathbf{BiPos}_\mathrm{f}$. Then $\mathbf{Retr}(\Lambda)$ has countable products.

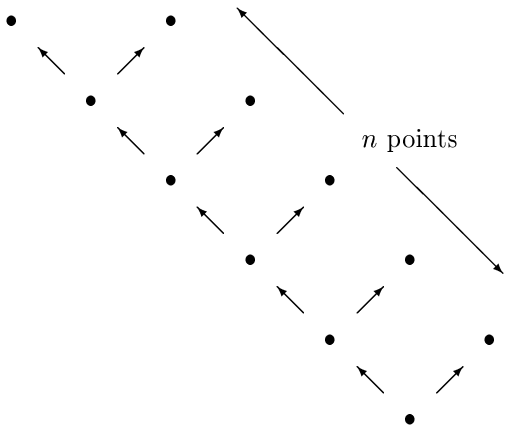


Figure 2.6.7: Lazy natural numbers

Proof Let $I = \omega_\perp^\top$ if Λ has top, $\omega_\perp$ otherwise, so $I \lhd \Lambda$. Let $X_i \lhd \Lambda$ for $i \in \omega$, *i.e.* $X_i \in \Lambda$ with $\mathsf{P}X_iX_i = X_i$. I claim $\prod_{i\in\omega} X_i$ is a retract of Λ^I, and hence of Λ, by

$$(p : I \to \Lambda) \quad \mapsto \quad \left[i \quad \mapsto \quad \begin{cases} \top & \text{if } i = \top \\ X_i(pi) & \text{if } i \in \omega \\ \perp & \text{if } i = \perp \end{cases} \right]$$

For any $(x_i : i \in \omega)$ may be extended to a unique $p : I \to \Lambda$ fixed by this, and any fixed point yields such a family. □

We have various kinds of "sums" by a similar trick.

$^+$§**2.6.9** It is not necessarily the case that the class of retracts of a domain is closed under bilimits.

Example Let X be the disjoint union of 2^n for $n \in \omega$, with a $\perp$ added. Then every finite lattice occurs as a retract of X, but no infinite one does. □

Here again we retreat to the case of **bcCont**.

Lemma Let $\mathcal{C}$ be a full subcategory of **bcCont** closed under retracts and (countable) products. Then $\mathcal{C}$ is closed under (countable) bilimits.

Proof Let $X : I \to \mathcal{C}^{cp}$ be a filtered diagram of comparisons; since we have retracts, w.l.o.g. the maps are embeddings. The proof really goes *via* the global (polymorphic) product, which we shall meet in Chapter V; in fact we show $\mathsf{bilim}_{i\in I} X_i \lhd \prod_{i\in I} X_i \lhd \prod_{i\in|I|} X_i$, by which we mean respectively the bilimit, the polymorphic product and the discrete product. Whereas in the last the terms in the family (x_i) may be arbitrary, in $\prod_{i\in I} X_i$ they must satisfy $x_i \le \pi_{ji}x_j$, and in the bilimit we have equality.

Thus we make $\prod_{i\in I} X_i \lhd \prod_{i\in|I|} X_i$ by

$$(x_i) \quad \mapsto \quad \left(\bigwedge \{\pi_{ji}x_j : i \le j\} \,;\, i \in I \right)$$

Continuity follows from directed distributivity, and the comparison condition holds because $\pi_{ii'}$ preserves $\bigwedge$ and then we are taking the inf of a larger set for i' than for i (where $i' \le i$); finally the operation preserves families which already satisfy the condition.

The comparison map $\mathsf{bilim}_{i\in I} X_i \to \prod_{i\in|I|} X_i$ which takes an element of the (bi)limit to the compatible family clearly factors through the inclusion of the polymorphic product. This inclusion is a retract simply by taking the directed sup:

$$(x_i) \in \prod_{i\in I} X_i \quad \mapsto \quad \bigvee\!\!\!^{\uparrow} x_i \in \mathsf{bilim}_{i\in I} X_i$$

Hence we have the required retract and $\mathsf{bilim}_{i\in I} X_i \in \mathcal{C}$. *cf.* Lemma 5.2.5. □

Proposition Let Λ be a β-model in $\mathbf{bcCont}_\omega$. Then $\mathbf{Retr}(\Lambda)$ is closed under countable bilimits, and is hence a small category of continuous domains. □

$^+$§**2.6.10** We now return to categories of domains to find saturated objects. The construction of this section is like $P\omega$ and depends on the foregoing results; that of the next section is more like D_∞ and follows on from proposition 2.6.4.

Proposition Let $\mathcal{C}$ be a small category of boundedly-complete continuous domains and Σ its (countable) class of finite objects. Then $\Lambda = \prod_{X\in\Sigma}(X^{\mathbb{N}})$ is a β-model and $\mathcal{C} \simeq \mathbf{Retr}(\Lambda)$.

Proof Since $\Lambda^{\mathbb{N}} \cong \Lambda$, the class of retracts of Λ satisfies the conditions of lemma 2.6.9 and so is closed under countable bilimits; since it contains Σ, it is $\mathsf{Bi}_\omega\,\Sigma \simeq \mathcal{C}$. On the other hand Λ^Λ is a bilimit of objects of the form X^Y with $X, Y \in \Sigma$, but these are in Σ since it's closed under exponentials, so $\Lambda^\Lambda \lhd \Lambda$ (somehow). □

$^+$§**2.6.11** We shall find a saturated domain for $\mathsf{Bi}_\omega\,\mathbf{Pos}_f$ by a more general technique which historically derives from the general construction of algebraic closures of fields.

Proposition $\mathsf{Bi}_\omega\,\mathbf{Pos}_f$ has a saturated domain.

Proof Let Σ be the category of finite posets with $\bot$ and embeddings. We seek a "saturated countable filtered colimit" of Σ.

Consider all finite strings of embeddings in Σ starting from the singleton, ordered by "amputation" (*i.e.* initial segments); these form a tree, T. There is a diagram $T \to \Sigma \subset \mathsf{Bi}_\omega\,\mathbf{Pos}_f^{em}$ given on points by taking the object at the end of the string and on arrows by the corresponding embedding.

Let Λ be the limit of this diagram (the name is justified since we shall have $\Lambda^\Lambda \lhd \Lambda$). By theorem 2.2.14 we construct it with pullbacks and bilimits and the maps in the limiting cone are projections.

Now let $X \in \mathsf{Bi}_\omega\,\mathbf{Pos}_f$. Then X is the bilimit of an ω-diagram of finite posets and embeddings, w.l.o.g. starting at the singleton. We therefore have a diagram $\omega \to \Sigma$, which may be extended to $\omega \to T \subset I$ by considering initial segments. This is a subdiagram of that defining Λ, so there is an embedding $X \rightarrowtail \Lambda$ as required. □

$^+$§**2.6.12** We sum up the results of this section for boundedly-complete continuous domains.

Theorem

(a) Let Λ be a β-model in $\mathbf{bcCont}_\omega$. Then $\mathbf{Retr}(\Lambda)$ is a small category of boundedly complete continuous domains.

(b) Let $\mathcal{C}$ be a small category of boundedly complete continuous domains. Then there is some $\Lambda \in \mathcal{C}$ with $\mathbf{Retr}(\Lambda) \simeq \mathcal{C}$.

(c) Any such category consists of retracts of countable bilimits of Σ-objects, where Σ is a class of finite boundedly-complete posets closed under retracts and products, and any such Σ will do. □

The corresponding results for $\mathbf{BiPos}_\mathrm{f}$ are weaker:

Theorem

(d) Let Λ be a β-model in **ContDom**. Then $\mathbf{Retr}(\Lambda)$ is closed under retracts, exponentials and countable products.

(e) $\mathsf{Bi}_\omega\,\mathbf{Pos}_f$ has a saturated object.

(f) Any small category of algebraic domains is of the form $\mathsf{Bi}_\omega\,\Sigma$ for Σ some class of finite posets closed under retracts and exponentials, and any such Σ will do. □

§2.6.6 suggests that we obtain nothing new by varying the weight of the category. We may improve the properties of Λ (*e.g.* Y is the *least* fixpoint combinator and $\Lambda \cong \Lambda \times \Lambda \cong \Lambda^\Lambda$) by using propositions 2.2.8 and 2.5.8.

Chapter 3

Type Polymorphism

3.1 Types are First-Class Citizens

§**3.1.1** Types first featured in programming languages in an *ad hoc* fashion as in FORTRAN, where there are "two kinds of numbers", and in BASIC and countless operating-system and applications command languages, which have numeric and string variables. Already here, however, we encounter *coercion* ("widening" in ALGOL 68), which is the automatic conversion of a value of one type (*e.g.* `INTEGER`) to the type of its context (*e.g.* `REAL`), and *overloading*, which is the use of the same symbol (*e.g.* +) for different operations depending on the types of its arguments (*e.g.* addition of numbers and concatenation of strings).

By the end of the 1960s, however, the possibility of extending the type structure was available to the programmer. Three features in particular may be recognised in ALGOL 68 (see [van Wijngaarden *et al.* 1975] or [McGettrick 1978]) or PASCAL [Jensen and Wirth 1975].

First, *enumerated types*, such as

$$week \quad = \quad \{Monday,\ Tuesday,\ Wednesday,\ Thursday,\ Friday,\ Saturday,\ Sunday\}$$

to replace the choice and passing of arbitrary numeric codes. This enforces a discipline on the programmer, and by explicitly restricting the choice of values of a parameter makes it more likely that its value will be correct, comprehensible to a future maintainer and consistent with subsequent extension. It also enables a clever compiler to optimise, performing say byte rather than word arithmetic on the value. Mathematically these of course correspond to finite coproducts.

Second, *records* or *structures* (corresponding mathematically to finite products) enable data to be organised (passed, copied, compared, ...) at a higher level than a number or string at a time. Finally, *variant records* provide for the type of an item of data to be varied, the type itself being saved in a *tag field*; mathematically this corresponds to an indexed sum. These features allow the implementation of operations to be *hidden* from the user, making it possible subsequently to modify it invisibly.

Languages which have such features are called *strongly typed*. Most of the *type checking* can be done at compile-time (as opposed to run-time) because it is syntactic and decidable.

§**3.1.2** Even with these extensions, we still have to duplicate code if we wish, say, to calculate the determinant of both real and complex matrices. To avoid this we now allow the type to be a variable, which phenomenon we call *polymorphism*.

It does not require very much of a mathematical background to realise that the determinant depends only on addition and multiplication, so this process may be performed with respect to any *ring*. Consequently one may define the determinant function parametrically on the base ring; in other words an additional argument is provided (which we shall by convention list first since the interpretation of subsequent arguments is dependent on it) specifying the type corresponding to the ring.

The code of the polymorphic determinant program, then, consists of the pure algorithm (either the defining polynomial with $n!$ terms in the coefficients or — more likely — Gaussian elimination with pivotting), into which are to be substituted the calls to the addition and multiplication subroutines appropriate to the kind of arithmetic in use. The algorithm, however, is given *uniformly* for all rings.

For the practical implementation of this, we need to pass to the program not just the *type* of the elements of the ring which will feature as coefficients in the matrix (which the compiler will use only for type-checking) but also the *operations* on the ring. We can, of course, already do this in FORTRAN: to the arguments we add functions $\mathbf{t}MULT(X,Y)$ and $\mathbf{t}ADD(X,Y)$, *calls* to which occur instead of $\times$ and $+$ in the code. More abstractly, the initial "type" argument to the routine consists of the entire signature, $(R,0,1,+,-,\times)$, of the ring.

It is the word "uniformly" which we shall pursue here. In this example it is precisely the same notion as the "natural" isomorphism of a vector space and its *double* dual which led Eilenberg and Mac Lane to their original formulation of category theory [1942]. We find that it leads naturally to the notion of *classifying topos*, which we take the opportunity to introduce (unfortunately, within the scope of the present work the hierarchy of categories of domains in §2.5 is the only occasion we find in the present work to make use of it).

In the particular example of the determinant there is in fact another hidden degree of polymorphism because (even fixing the base ring as $\mathbb{R}$) there are matrices of different sizes. Of course this is because we are really dealing with polymorphism over the category of $\mathbb{R}$-modules. When we discuss this example we shall therefore stick to 2×2 matrices and consider just the function $a\times d - b\times c$ since this suffices to make the point. However it may dawn on us that the design of the "conceptually perfect" programming language is inevitably going to involve the whole of modern Algebra.

§**3.1.3** Unfortunately not all of the type constructions one might wish to consider in a programming language are functorial in their parameters. The foremost example of this is the exponential or function space which, like the *single* dual of a vector space, is *contra*variant in its first argument. In such circumstances it would appear that this interpretation breaks down.

A successful but nevertheless *ad hoc* attempt to modify the category of types in order to render the function space construction *co*variant in the first argument has already been discussed in chapter II. We shall return to it in chapter V where we motivate the claim that it is an example of what we mean by a *continuous type dependence*.

This leads us to consider languages with type-valued expressions. These may have free variables either of some ordinary type(s) — for instance the type of solutions of some parametric equations — or of type *type* — *e.g.* the function space.

In such languages it may no longer be possible to perform type checking at compile-time (clearly, for instance, we cannot know whether an expression has type the solution-set of an equation until we know its value). Indeed it may even be the case that type checking is undecidable, as happens with the language we introduce in §3.5.6.

§**3.1.4** There is another notion of polymorphism which predates the advent of electronic computers. This is the belief that, say, the identity function is really unique rather than occurring once for each type. We say that the type of $\lambda x.x$ is really "$\forall X.X\to X''$. We shall introduce two languages, the *second order polymorphic lambda calculus* and *PONDER*, which explore this idea. PONDER admits of a variety of curious constructions which one might well consider to have no business to exist.

These will, however, be presented late in the chapter, after we have introduced indexed category theory and applied it to natural deduction, since there are important formal parallels between the constructs.

These are *second-order*, *untyped* languages in the sense that the types are terms of a certain single higher type which is not itself a type of the language. It is therefore analogous to the large universe of sets. In the determinant example we were interested in types with the signature of rings rather than sets, whereas here we have only one kind of type. Additionally, it is restricted to the second order in that we do not have facility to consider collectively *all* (say) algebraic theories.

It would be possible to construct a more elaborate category of types in which the ordinary types have mixed in some "higher" types constructed as powers of the external type of types. There is of course a strong tradition in set theory of such stratification but this approach rapidly leads to great complexity when the objects we want to consider consist not simply of types or elements but algebraic structures on a type together with chosen elements.

We shall return to this topic in §3.5.11.

§**3.1.5** This kind of approach also necessarily excludes the compiler itself from being well-typed, since it has to comprehend *all* types *whatever*.

Consequently we shall follow what has in the past frequently been a dangerous path and seek models in which there is an "ordinary" type of *all* types. We are saved from contradiction because the logic interpretable in our system is extremely weak (as we showed in chapter I, if we ask for fixpoints we lose most of Comprehension and all Negation). Nevertheless when, in chapter V, we do construct a model, we still find ourselves sailing very close to the wind.

§**3.1.6** There is a strong formal connection between (polymorphic) type theory and natural deduction or proof theory, roughly as indicated in the following table:

Proposition	Type
Deduction	Term or Function
Proof	Element
Conjunction and True	Products and Singleton
Disjunction and False	Coproducts and Empty
Implication	Function-space
Substitution	Substitution
Existential Quantifier	Sum or Coproduct
Universal Quantifier	Product

This may be extended to programs, but the novelty of this attitude means that it is difficult to identify the correct words to use in the table.

We shall take advantage of this analogy by exploring the case of natural deduction first. Since it involves lattices rather than categories, its features can be identified with less work. It nevertheless shows the importance of pullback functors and their adjoints.

§**3.1.7** The final ingredient of this present chapter is a discussion of the rôle of variables. The need for such an explicit treatment arises from the fact that since we have both type and term variables we shall be making more sophisticated use of them than we have done so far. Such a discussion also throws further light on the use of indexed categories.

The way in which mathematicians in general use variables is mysterious (in both the medieval and modern senses of the word) to the general population, as one quickly discovers if one attempts to teach third-form algebra. Consider for example the following quotations from a typical school textbook:

(a) "Put $x = 3$, $y = 7$ in $4x + 2y$"

(b) "Simplify $-5a + b^2 - a(b - a) + a - b$"

(c) "Factorise $x^2 + 5x + 6$"

(d) "Solve $x^2 + x - 2 = 0$"

(e) "Solve $x^2 - 41y^2 = 1$"

(f) "Solve $\sqrt{x+4} + \sqrt{x-1} = 5$"

(g) "Sketch $x^3 + 3xy = 2$"

(h) "Prove $(x^2 + y^2)^2 - (x^2 - y^2)^2 = 4x^2y^2$"

(i) "Integrate $\int \frac{dx}{\sqrt{a^2+x^2}}$"

(j) "Find $\int_0^1 \frac{dx}{\sqrt{1-x^2}}$"

Can we be surprised that most people throw up their hands in horror at the sight of Algebra, having been brought up on this diet of philosophical confusion?

§**3.1.8** In both programming and pedagogy, variables are introduced as having values in an "environment". This is a reasonable understanding of x and y in example (a), and — at a stretch — of a and b in (b) if the pupil is persuaded to read *apples* and *bananas* (what is the square of a banana?). The view is more tenable in programming because a compiler will demand that all variables be *declared*, which is tantamount to being given (or at least having) a value.

But clearly this breaks down in the remaining examples from §3.1.7, especially those involving calculus. In (d) and (e) the symbols x and y are better described as *indeterminates* (there is an additional confusion in that (e) is a Diophantine equation: there is a contextual understanding that we seek all *integer* solutions). The process of dealing with such problems raises another philosophical difficulty which we expect schoolchildren to grasp without explanation. Historically this was described as *analysis* (in which we treat the indeterminate as if it were known and deduce its value) and *synthesis* (in which we make use of this knowledge to prove the solution).

Example (f) particularly illustrates analysis and synthesis; we solve it by twice rearranging and squaring to deduce $x = 5$. However had we put -5 on the right the typical schoolchild would have made the same deduction, though in this case there is no solution since $\sqrt{}$ by convention takes nonnegative values. The *analysis* deduced from the assumption that the value exists what it must be, then *synthesis* determines whether it does exist.

In examples (b), (c) and (h) the variables are intended to stand for *generic* numbers. We do *not* mean that these equations hold *for each value which one might care to substitute*, but that they hold *formally* and may be manipulated as such. We see this again when school textbooks talk of "the function $\mathsf{sin}\, x$", which notation we improve to $x \mapsto f(x)$ or $\lambda x.a$. In the field $\mathbb{F}_5$ of order 5 the polynomial $x^5 - x$ always takes the value 0, but $x^5 - x = 0$ is not an identity in this sense.

The relationship between the *environmental* and *generic* use of variables is the same as that between models and theories. If there are *enough* values for the variable to take (as there are not in $\mathbb{F}_5$) then we have a *completeness theorem*, that a proposition which holds in all environments is provable.

In the case of an *algebraic* theory (such as rings or combinatory algebra), variables may be treated as generators for a free algebra. For such a theory completeness is trivial, since provability coincides with truth in the free algebra. We shall touch briefly on coherent logic and find a *generic* model (in the classifying topos) for which the same is true.

§**3.1.9** The typical environmental approach to models of the λ-calculus (*e.g.* [Meyer 1982]) or to model theory presupposes a fixed infinity of variables and all possible assignments of values to them. When we try to treat variable binding we then have essentially to "overwrite" the value-assignments, and we are obliged to imagine some process for picking new variables out of the air (*i.e.* which do not occur freely in some collection of expressions). Of course this process is not an *algorithm* because, of its very nature, it cannot be internalised.

Proposition In $\Lambda[x_0, ..., x_{k-1}]$ there is a term $\#$ such that $\#\mathbf{n}$ enumerates the terms as $\mathbf{n}$ runs through (say, Church's) numerals.

Proof (sketch) Let π_0, π_1 be the unpairing functions arising from the cross-diagonal bijection $\mathbb{N} \times \mathbb{N} \cong \mathbb{N}$ (§1.4.2); these are primitive recursive and hence λ-definable (§1.1.7). Now let

$$\#\mathbf{n} \quad = \quad \begin{cases} x_i & \text{if } \pi_0\mathbf{n} \equiv i \bmod k+1 \\ \#(\pi_0(\pi_1\mathbf{n}))(\#(\pi_1(\pi_1\mathbf{n}))) & \text{if } \pi_0\mathbf{n} \equiv -1 \bmod k+1 \end{cases}$$

which is also primitive recursive. Then by proposition 1.1.14, $\Lambda[\vec{x}] = \{\#\mathbf{n} : n \in \mathbb{N}\}$. Application and abstraction are also recursive, so there exist $!$, $\clubsuit_i$ with $\#(!\mathbf{mn}) = \#\mathbf{m}(\#\mathbf{n})$ and $\#(\clubsuit\mathbf{m}) = \lambda x_i.\#\mathbf{m}$ for all $m, n \in \mathbb{N}$. $\square$

This does not, of course, work for infinitely many variables.

§**3.1.10** The λ-calculus and predicate logic are syntactically more complicated than algebra in that they have *variable binding*. This also features in examples (i) and (j) of §3.1.7, where it is familiar that the variable of integration is "dummy". Renaming of dummy variables is called the **α-rule**.

The fact that the same *identifier* may occur both free and bound, and indeed may occur bound more than once, is another source of confusion, with the effect that one must apply a modicum of thought to the algorithm of substituting an expression for a free variable. There is also a distinction between an *identifier* and a *name*, the latter being a particular occurrence.

The *range* of a name is the subterm delimited by its binding or declaration, or the whole term if it is free. Since subterms are nested (not overlapping), so are ranges. The *scope* of a name is its range minus any nested range of a name with the same identifier; thus within its scope a name is determined by the identifier.

There are syntactic devices for overcoming this confusion, by eliminating bound variables altogether. The transition from λ-calculus to combinatory algebra which we made in chapter I is the prime example of this. Another technique is to replace each use of a bound variable by a numeral, given by the number of binding operators which separate the use from the binding operator which governs it. Arguably the almost universal practice amongst programming languages of identifying subroutine arguments (bound variables) with their values under application *positionally* already relegates them to mere names of product projections.

It is usual for a compiler to resolve these things and build its *symbol table* at a very early stage in its processing. It is well known that a *context-free grammar* cannot recognise repetitions of a subterm; for this and other reasons the syntactic phase of compilation is devided into *tokenisation* (in which we strip spaces and comments and recognise identifiers and operators as syntactic atoms using a *regular grammar*) and *parsing* (in which the structure of the program is found).

For a practical discussion of compilers see [Aho and Ullman 1977] and its bibliography.

§**3.1.11** Yet another confusion which arises with variables is the old and thorny problem of *use* (value) and *mention* (name). This is particularly dangerous in the context of assignment: when we write "$x := y$" we *use* y and *mention* x.

The authors of ALGOL 68 put a great deal of effort into assignment, making a novel use of types (or, as they idiosyncratically called them, *assignation* and *modes*). The signature of the assignment (say for integers) is $(\mathbf{refint}, \mathbf{int})$, and if y happens to have mode **refint** we have first to coerce (dereference) it to **int**.

§**3.1.12** There is another problem which arises when a named procedure uses a variable which is not defined within the procedure but (possibly) within an enclosing one, and this procedure is called in a different context in which the variable is also defined. This is called the *dynamic free variable problem*. Which value for this variable should be used?

§**3.1.13** The final problem with variables which we consider arises in Natural Deduction from the possibility of an empty domain. From the two rules

$$\frac{\forall x.\phi(x)}{\phi(t)}\ (\forall\mathcal{E}) \qquad (\exists\mathcal{I})\ \frac{\phi(t)}{\exists x.\phi(x)}$$

it is an easy matter to deduce a contradiction. There has been much discussion in the Model-theoretic literature — both classical and categorical — of how this problem is to be resolved, with a suggestion (perhaps applicable to law or politics) even that deduction be not assumed transitive! A more serious proposal, due to Mike Fourman, is the use of partial elements.

§**3.1.14** The aim of this chapter, then, is the introduction of indexed category theory as a means of presenting notions of polymorphism and clearing up many of these confusions with variables.

We begin by considering presentations of syntax in terms of indexed "categories" and point out the significance of structure such as products and functors. These ideas are applied to arithmetic, context-free grammars, algebra, λ-calculus and finally natural deduction, with particular reference to the interpretation of the variable binding operators. We also remark that 2-categories and enriched categories can be employed for the same purpose.

In section 3 we consider the determinant as an example of a generic (uniform, natural) construction. It gives rise to a natural transformation between functors $\mathbf{Rng} \to \mathbf{Set}$. We introduce classifying toposes and sketch their construction for coherent theories. Then we discuss stronger forms of logic which happen "accidentally" in the topos and indicate under what conditions these are preserved. Finally we show how to make an indexed category which codes the language of geometric logic.

In section 4 we look at the second order polymorphic λ-calculus and Fairbairn's language PONDER, demonstrating that the "type quantifier" $\forall$ is right adjoint to substitution.

Finally section 5 concerns strong forms of polymorphism including the type of types, showing how recursion and type-of-types fit into the indexed formulation of polymorphism.

In chapter V we shall construct models for these languages.

3.2 Semantics of Variables

§**3.2.1** Let us consider what we would like to be able to do with variables anyway, taking the line that they denote "generic" typed values. We expect simple categorical descriptions of substitution and variable binding operators, and the α rule to fall out as a triviality.

The operations on variables are

(i) *proliferation*: extension of the class of variables in which a given term is to be defined,

(ii) *substitution*,

(iii) *binding* by λ, $\forall$, $\exists$, *etc.*,

(iv) *α-rule*: transparent renaming of dummy (bound) variables,

(v) creation of new variables when required.

§**3.2.2** The basic idea is to "stratify" or "index" formulae with respect to the variables which are (potentially) free in them. Each stratum or *fibre* is named by a string of variables of certain types. Alternatively we may forget (the names of) the variables and list instead the types; it will become apparent that this list is in fact the *product* of the types in it.

Note This depends on being able to forget and duplicate variables, *cf.* §3.2.20.

The substance of each stratum is the "free" or "term" model of the language. It is generated by the variables in the list and carries the "algebraic" structure corresponding to the operations in the language. Alternatively we may think of it as the "universe of all definable structure" derived from the variables as "generic" values.

§**3.2.3** What about substitution, say $a[x := u]$? It seems to be appropriate to regard as primitive the operation of *simultaneously* substituting for *all* of the free variables of a. If we wish to substitute for only some of them, we make the identity substitution in the remaining places. [There is an argument to suggest that individual substitution is a strictly more powerful operation: §3.2.20.]

Suppose, then, that x is the only free variable of a, whilst u has its free variables amongst $\vec{y} = y_1, y_2, ..., y_m$. The operator $[x := u]$ is then a map from the fibre over x to that over $\vec{y}$. Since substitution is meant to respect the operations in the language, this map is a *homomorphism* of those operations.

§**3.2.4** If a has not just one free variable, x, but a string of them, $\vec{x}$, we must make a (simultaneous) multiple substitution $[\vec{x} := \vec{u}]$ for them, and all substitutions are of this form.

Moreover if $\vec{u}$ have free variables $\vec{y}$, the substitutions which may be performed after (composed with) $[\vec{x} := \vec{u}]$ are those of the form $[\vec{y} := \vec{v}]$, and the resultant substitution is

$$[\vec{x} := \vec{u}][\vec{y} := \vec{v}] = [\vec{x} := \vec{u}[\vec{y} := \vec{v}]]$$

It is then easy to see that we have a *category* of substitutions, whose *objects* are the strings of variables (or their types), whose *morphisms* are the substitutions, whose *composition* is that just given and whose *identity* is $[\vec{x} := \vec{x}]$.

Clearly the substitutions are determined by the strings, $\vec{u}$, of terms. These strings have type strings the same as those of the corresponding $\vec{x}$.

Proposition The category of terms (§1.3.10) is dual to the category of substitutions. It has finite products, corresponding to listing of types. Each morphism gives rise contravariantly to a homomorphism between the corresponding fibres, and every such homomorphism arises in this way. □

We refer to the category of terms as the *base category* and denote it by $\mathcal{B}$. The fibres are the "concrete" or *regular representation* of $\mathcal{B}$, *i.e.* just the hom sets $\mathcal{B}(-,-)$. $\mathcal{B}(X,Y)$ is the collection of terms of type Y in the fibre over X.

§**3.2.5** Now let's look at the sense in which x, *quâ* term, is generic in the fibre over itself, *quâ* variable. Let a be any term, say with free variables amongst $\vec{y} = y_1, ..., y_m$. We may form the substitution $[x := a]$, which corresponds to a morphism from $\vec{y}$ to x and gives rise to a homomorphism from the fibre over x to that over $\vec{y}$. This is the *unique* homomorphism taking x to a.

Definition A point x in the fibre over $X \in \mathcal{B}$ is said to be *generic* of type X if for any a of type X in the fibre over Y there is a unique $\alpha : Y \to X$ in $\mathcal{B}$ such that the corresponding homomorphism from the fibre over x to that over Y takes x to a. We write $\ulcorner a \urcorner$ for α.

§**3.2.6** Most of the rest of this section is concerned with examples of this structure; observe that as yet we have avoided being specific about the nature of the fibres.

The most familiar example is that of polynomials, for which the language is that of *commutative rings*, consisting of *addition*, *multiplication*, *etc.* The base category simply counts the number of variables. The fibre over $\vec{x} = x_1, ..., x_n$ is the free ring on these generators and the substitution homomorphisms are the obvious evaluation maps; these are indeed precisely all of the homomorphisms between these rings.

There's something a little peculiar here about restricting attention to *free* rings. It would seem to be appropriate to extend the base category to the opposite of the category of *all* commutative rings, or perhaps to the category of algebraic varieties. This of course leads us into algebraic geometry.

§**3.2.7** Let's look at the preceding example from the point of view of universal algebra. An object of the base category is, to all intents and purposes, just a *number* (of variables), and the product of objects corresponds to *addition* of numbers. Since every object is a product of copies of a certain single object, we need only consider morphisms into that, and from $\vec{x}$ these are all expressions in the free variables $\vec{x}$.

The indexed algebra described above is easily seen to be generated by the defining operations of the algebraic theory. However it is sometimes convenient to add to that theory *all* derived operations obtained by substituting one within another and with them the corresponding equations which say when an operation is obtained in this fashion. For example for groups we specify not only *nullary* (identity) and *binary* multiplication but also the product of one, three, four, ... elements. The closure of the signature (system of operations) under substitution in this fashion is called the *clone* of the theory.

The clone of the theory of rings is just the collection of all polynomials in however many variables. More generally it gives all the morphisms into x of the base category. The entire base category $\mathcal{B}$ is called the *Lawvere presentation* of the theory.

Proposition

(a) The objects of the category $\mathcal{B}$ arising from a finitary algebraic theory in this way are in bijection with $\mathbb{N}$, the natural numbers, and the product of two objects is given by the sum of the corresponding numbers. Any category of this form arises uniquely (up to equivalence) from some finitary algebraic theory.

(b) The category of models of an algebraic theory is equivalent to the category of functors $\mathcal{B} \to \mathbf{Set}$ which preserve finite products, and natural transformations between them.

(c) $\mathcal{B}^{op}$ is the category of finitely generated free algebras and homomorphisms. □

§**3.2.8** The passage from single-sorted to many-sorted algebraic theories in this presentation is easy and natural: it is given simply by dropping the condition that $\mathsf{ob}\,\mathcal{B} \cong \mathbb{N}$. For the extension to essentially algebraic theories, we need to make $\mathcal{B}$ a *left-exact* or *lex* category, *i.e.* one with all finite limits (see §1.2.13). The "essentially algebraic" theory of categories yields an example of this. This is usually presented as a two-sorted theory ("objects", C_0, and "morphisms", C_1) with three total operations ("identity", "domain" and "codomain") and one partial ("composition").

The domain of the composition function is not the whole of $C_1 \times C_1$ but that subobject carved out as the equaliser of $\pi_0 ; \mathsf{cod}$ and $\pi_1 ; \mathsf{dom}$. It is the set of "composable pairs" and is written C_2. It may also be constructed by means of either of the following pullback diagrams. The first is the one more commonly seen, but the second will prove more useful to us in chapter V because $\langle \mathsf{dom}, \mathsf{cod} \rangle : C_1 \to C_0 \times C_0$ is a *display map*, which is to say the inverse image of $\langle x, y \rangle \in C_0 \times C_0$ is to be a *type*, $\mathcal{C}(x, y)$, viz. the set of maps $x \to y$ in $\mathcal{C}$; this will ensure that we can indeed form the pullback.

$$
\begin{tikzcd}
C_2 \arrow[r, "\pi_0"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & C_1 \arrow[r, "\mathsf{dom}"] \arrow[d, "\mathsf{cod}"] & C_0 \\
C_1 \arrow[r, "\mathsf{dom}"] \arrow[d, "\mathsf{cod}"'] & C_0 & \\
C_0 & &
\end{tikzcd}
\qquad
\begin{tikzcd}
C_2 \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & C_1 \times C_0 \arrow[d, "\langle d, c\rangle \times 1"'] \\
C_0 \times C_1 \arrow[r, "1 \times \langle d, c\rangle"'] & C_0 \times C_0 \times C_0
\end{tikzcd}
$$

Figure 3.2.8: The definition of an internal category

Proposition $\mathcal{B}^{op}$ is the category of finitely *presented* algebras and homomorphisms (*cf.* proposition 3.2.7c). □

§**3.2.9** The presentation of a many-sorted finitary essentially algebraic theory is then just a left-exact category $\mathcal{B}$. Its category of models is $\mathbf{LEX}(\mathcal{B}, \mathbf{Set})$, the category of finite limit preserving functors from $\mathcal{B}$ to **Set**, and natural transformations between them.

There's nothing special here about the category **Set**: we may interpret an essentially algebraic theory in any category $\mathcal{E}$ with finite limits, and the category of models is $\mathbf{LEX}(\mathcal{B}, \mathcal{E})$. For single-sorted algebraic theories this is familiar, writing $\mathbf{G}$ for the Lawvere theory of groups, the category of topological groups is $\mathbf{LEX}(\mathbf{G}, \mathbf{Sp})$. $\mathcal{E}$ will commonly be a topos.

However for $\mathcal{E}$ we may instead put another algebraic theory; a left-exact functor $\mathcal{B} \to \mathcal{E}$ is then an *interpretation* of one theory in another, that is, an expression of the operations of (the theory presenting) $\mathcal{B}$ in terms of those of $\mathcal{E}$. The following example is an instance of this.

$^{+}$§**3.2.10** Context free grammars may be treated as a special case of a many sorted finitary algebraic theory. Recall that a *concrete context-free grammar* over an *alphabet* A consists of a collection of *syntactic sorts* (we prefer this to the usual *syntactic category* for the obvious reason) and *production rules*. For example decimal integer arithmetic expressions may be described by

$$\mathrm{A} = \{0,\ 1,\ \ldots\ 9,\ +,\ -,\ \times,\ \div,\ (,\ )\}$$

together with

$$
\begin{array}{rcl}
D & : & := 0 \mid 1 \mid \ldots \mid 9 \\
N & : & := D \mid ND \\
I & : & := N \mid +N \mid -N \mid (E) \\
T & : & := I \mid T \times I \mid T \div I \\
E & : & := T \mid E + T \mid E - T
\end{array}
$$

It is an easy matter to derive from this a five-sorted algebraic theory with ten constants, seven unary operations, five binary operations and no equations.

It would be more accurate to describe this not as a single theory, but as an interpretation of one abstract theory (of arithmetic) in another (words in A). As lex categories, an interpretation of theories is given by a lex functor.

Proposition Let Γ be a context-free grammar in an alphabet A. Then there are naturally arising categories with products, $\mathsf{\Gamma}$ and $\mathbf{A}$, and a product-preserving functor $\mathsf{\Gamma} \to \mathbf{A}$. Moreover this functor is faithful iff Γ is unambiguous.

Proof The *objects* of Γ are strings of syntactic sorts, $\vec{X}$. The *morphisms* from $\vec{X}$ to $\vec{Y}$ are strings of productions of sorts $\vec{Y}$ with free syntactic variables of sorts $\vec{X}$ together with their *detailed* parsing forests; the terms are at the roots of the trees, the nodes are production rules and the leaves are syntactic variables. The *identity* is as usual given by variables. *Composition* is given by substituting trees for leaves in parsing forests. Γ is therefore the *free* category with products on the production rules (*quâ* morphisms).

The *objects* of $\mathbf{A}$ are natural numbers n (or finite sets of *any* syntactic variables irrespective of sort). *Morphisms* $n \to m$ are m-strings of words in A and n syntactic variables (without any kind of parsing information). *Identity* and *composition* are by variables and substitution. Notice that we have no reference to "words in A" as the "free monoid on A"; this is because that composition operation is concatenation, which is a red herring as far as parsing is concerned.

The *functor* $\Gamma \to \mathbf{A}$ gives the length of the string on objects and the term being parsed (with its free variables) on morphisms, *i.e.* it throws away the parsing information. It is faithful iff whenever there are two parsing trees for the same term (in the same variables) then these coincide. □

§**3.2.11** Simply by choosing some syntactic representation of an algebraic theory it is easy to give an interpretation of it in terms of an *abstract* context-free grammar. However it is necessarily abstract, since a *concrete* interpretation (by words in an alphabet) would implicitly be a solution of the word problem for the theory, *i.e.* the determination of whether two expressions in given generators and relations are provably equal in the theory, and this is known not to be possible.

Consideration of the analogy between context free grammars and many sorted algebraic theories is enlightenling. As a fairly trivial point, we notice that brackets (such as in the rule for I in §3.2.10) serve as inverses to the inclusion of one sort in another.

The syntactic sorts of course correspond to the algebraic sorts, and when they occur on the right hand side of a production rule they correspond to variables. We might describe them as *syntactic metavariables*, since in the context of a programming language there will also be *semantic* (*object*) *variables*. Some languages, notably *C* [Richie *et al.* 1975], have "macro" facilities, allowing parametrised substitution into the source text *before* it is processed by the compiler; here the analogy between the two kinds of variables is apparent.

§**3.2.12** None of the foregoing examples has involved variable binding operators, so let us now turn our attention to the λ-calculus. Because of the equivalence with combinatory algebra (chapter I) we already know what the structure of the fibres and substitution maps must be: it remains only to investigate the interpretation of λ itself.

We may as well deal with the *typed* λ-calculus. As in §3.2.2 the objects of the base category are strings of variables of various types, and the morphisms are strings of terms whose free variables are as listed by the domain and whose types are given by the domain; this is precisely the construction of §1.3.10. The fibres are also the sets of all terms in the given variables, identified under β-equivalence; they are organised into types within the fibres, and substitution respects type.

Application is definable structure, being defined between terms of types $A \to B$ and A with result B; it is preserved by substitution. As with algebraic theories the fibre is an algebra for the operation; it is free if we impose Curry's equations (§1.1.14). There is a *generic* instance of application with these types, namely $f \bullet a = fa$ in the fibre over $(A \to B) \times A$.

§**3.2.13** Let us write $[\vec{x}\,;A]$ for the type A in the fibre with free variables $\vec{x}$. Abstraction is then a function $\lambda y : [\vec{x}, y\,;A] \to [\vec{x}\,;Y \to A]$ where Y is the type of y.

The β-rule is the means by which we relate the *internal* notion of application to the *external* notion of substitution. Since we may separately substitute as we please for y, it suffices to consider the case $(\lambda y.a)y = a$. We have to proliferate $[\vec{x}\,;Y \to A]$ and $[y\,;y]$ to the fibre over $(\vec{x}, y)$; it takes the full force of proliferated substitution to define a map

$$[\vec{x}, y\,;A] \to [\vec{x}\,;Y \to A] \to [\vec{x}, y\,;Y \to A] \times [\vec{x}, y\,;Y] \to [\vec{x}, y\,;A]$$

which the β-rule requires to be the identity.

Looking at this another way, we have defined a post-inverse to λy. The *typed* η-rule says that this is also pre-inverse. This is of course what we did in §1.3.10.

Since in this presentation $[\vec{x};Y]$ is just the hom-set $\mathcal{B}(X,Y)$, these two rules together formulate the fact that $\mathcal{B}$ is cartesian closed. Observe, however, that the *fact* that $\mathcal{B}$ is cartesian closed may be formulated equationally, and hence in a world (instead of **Set**) which merely has finite limits.

The untyped λ-calculus (with self-application and fixpoints) requires further structure in $\mathcal{B}$. The remark about the strength of proliferated substitution will be amplified in §3.2.20 when we mention alternative presentations.

§**3.2.14** In all of the foregoing examples the fibres have been *discrete* sets, and have in fact carried no structure which was not already present in (the hom-sets of) the base category. For the application to polymorphism (and before that to natural deduction and type theory), the fibres acquire the structure of a *category* and the substitution homomorphisms are *functors*. The remarkable thing is that quantifiers, sums and products turn out to be their *adjoints*.

Let us suppose we have a First Order theory $\mathcal{T}$ in a language with certain sorts, operations, *etc.* Consider the language of the first order formulae, possibly including variables of the respective sorts, together with the connectives $(\top, \wedge, \bot, \vee, \rightarrow, \forall, \exists)$ and the relation $\vdash$ of $\mathcal{T}$-provability. We shall make this into an indexed structure (called a *hyperdoctrine*).

Since we now have a *relation* as well as operations, we no longer have a discrete set in the fibres. In fact in general we may build a 2-category, whose arrows are the *methods* (not just the *fact*) of proof that $\phi \vdash \psi$ and whose 2-arrows are the proof transformations. Seely [1977] does this, but the extent of his detail is far beyond what is appropriate to demonstrate the equivalence between *categorical* and *proof-theoretic* notions, so we shall restrict attention to the provability *preorder*.

§**3.2.15** The base category, as before, gives the *type* and *term* structure of the object language. The fibre over each string of types is the *Lindenbaum algebra* of propositions possibly containing free variables of the given types. This carries a Heyting algebra structure, with definable operations $\top$, $\wedge$, $\bot$, $\vee$ and $\Rightarrow$.

We pick just three rules from natural deduction to show that the connectives are interpreted by products, coproducts and exponentials; the last of these is historically known as the *modus ponens*.

$$\frac{\phi \vdash \theta \quad \psi \vdash \theta}{\phi \vee \psi \vdash \theta}\ (\vee\mathcal{E}) \qquad \frac{\chi \vdash \phi \quad \chi \vdash \psi}{\chi \vdash \phi \wedge \psi}\ (\wedge\mathcal{I}) \qquad \frac{\chi \vdash \phi \quad \chi \vdash \phi \Rightarrow \psi}{\chi \vdash \psi}\ (\Rightarrow \mathcal{E})$$

Classically we have the $(\neg\neg\mathcal{E})$ rule, making this a Boolean algebra, but (to a large extent because there is no analogue of this for categories) it is more appropriate to drop this in favour of *intuitionistic* logic.

Since the connectives are definable structure, as in the algebraic case they are preserved by the substitution functors. In the proper-category case we have a cartesian closed category with finite limits and colimits, this structure being preserved by substitution maps.

§**3.2.16** The crucial observation, due to Lawvere [1969], is that the rules for the quantifiers state that they are *adjoints to substitution*. This is illustrated by the two rules

$$\frac{\psi \vdash \phi}{\exists x.\psi \vdash \phi}\ (\exists\mathcal{I}) \qquad (\forall\mathcal{E})\ \frac{\phi \vdash \forall x.\psi}{\phi \vdash \quad \psi}$$

(where x does not occur freely in ϕ). These may be read directly as the adjunction between the functor (written, as usual, in invisible ink) which proliferates the free variable x and respectively

$\forall x$ on the right and $\exists x$ on the left. [The Frobenis law, $[\exists x.(\phi(x) \wedge \psi)] = [(\exists x.\phi(x)) \wedge \psi]$, is also needed to characterise the quantifiers.]

These proliferation functors are substitutions over product projections in the base category (the relevant terms being simply variables *quâ* terms in (more) free variables). What do quantifiers over general morphisms mean? They in fact turn out to be generalised to forms "$\exists x.\phi(x) \wedge \psi(x)$" and "$\forall x.\phi(x) \Rightarrow \psi(x)$", which are already familiar from mathematical idiom. (In fact the reader may already have spotted the notations "$(\forall x : \phi(x))(\psi(x))$" or sometimes "$\forall x : \phi.\psi$" and "$(\exists x : \phi(x))(\psi(x))$" or "$\exists x : \phi.\psi$" in this work: these have been used in recognition of this.)

$$\begin{tikzcd} X \times Y \arrow[d, "\pi_1"'] & \{(x,y) : \phi(x,y)\} \arrow[d, mapsto, "\pi_1"'] \\ Y & y \end{tikzcd}$$

We shall see more of (adjoints to) substitution over arbitrary base maps when we discuss the indexed presentation of **Set** in the next chapter.

§**3.2.17** There is a technicality in this called the *Beck condition*. [Indeed there are other important examples of indexed categories with adjoints to substitution, and the Beck condition sems to characterise those we call logical.] We want to be sure that substitution and quantification interact properly, in the sense that it doesn't matter whether we proliferate variables before or after quantifying. More generally, we want to be able to make substitutions inside quantifiers (though not, of course, for the variable being bound). This is expressed categorically by requiring that if the left-hand figure is a pullback in the base category then the right-hand square must commute (at least up to isomorphism):

$$\begin{tikzcd} C \times_A B \arrow[r, "\gamma"] \arrow[d, "\delta"'] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "\beta"'] \\ B \arrow[r, "\alpha"] & A \end{tikzcd} \qquad \begin{tikzcd} \mathsf{P}(C \times_A B) \arrow[r, "\mathsf{P}\gamma"] & \mathsf{P}C \\ \mathsf{P}B \arrow[r, "\mathsf{P}\alpha"'] \arrow[u, "\forall\delta"] & \mathsf{P}A \arrow[u, "\forall\beta"'] \end{tikzcd}$$

We also need a Beck condition for $\exists$. However this follows automatically from the fact that a diagram of left adjoints commutes iff the corresponding diagram of right adjoints does (so long as they both exist, of course).

§**3.2.18** The structure we have built so far, in which

(i) the base category has finite limits,

(ii) the fibres have finite limits, finite colimits and exponentials,

(iii) the substitution functors preserve the structure of the fibres

and

(iv) have adjoints on both sides satisfying the Beck condition is called a *hyperdoctrine*. We shall apply this to **Set** in the next chapter, and (after deleting equalisers and colimits) to categories of domains in chapter V.

The really basic example of a hyperdoctrine is the sub-object structure on a category of sets (topos). In this case substitution over α is just the inverse image α^{-1} and its adjoints are quantification. The Beck condition holds automatically in this case. We may regard an arbitrary hyperdoctrine as a structure which *forces* the subobject structure on (a category containing) its base category, in the following sense:

Proposition Let $\mathcal{P} \to \mathcal{C}$ be a hyperdoctrine. Then there is a category $\mathcal{S}$ whose subobject-structure yields a hyperdoctrine and in which $\mathcal{C}$ is fully embedded such that $\mathsf{Sub}_{\mathcal{S}}(X) \cong \mathtt{P}X$ for $X \in \mathcal{C}$.

Proof (sketch) The *objects* of $\mathcal{S}$ are "models of equality", *i.e.* objects $X \in \mathcal{C}$ together with $\mathcal{P}$-equivalence relations, $(=_X) \in \mathtt{P}X$ (satisfying the usual equations). The *morphisms* are from $(X, =_X)$ to $(Y, =_Y)$ are $\mathcal{P}$-relations (objects of $\mathtt{P}(X \times Y)$) which are functional, total and respect equality. $\square$

In the proof of this result we use the Beck condition in showing that relational (and hence functional) composition is associative (as Gavin Wraith has pointed out to me). This result shows that the Lawvere dictum (that quantification is adjoint to substitution) should be qualified by saying that the Beck condition characterises *logical* applications. Seely [1983] gives further discussion, claiming that the Beck condition for a particular pullback in the base says that "the category *knows* that this is a pullback". When we discuss indexed domain theory in Chapter V we shall encounter indexed categories whose substitution maps have left adjoints which do not satisfy the Beck condition. Other examples arise in Universal Algebra, for instance the indexing of categories of modules over rings, or algebras over theories.

We may take this one stage further and *force* the higher-order logic. [In other woords, we can build a category in which $\exists$ and $\forall$ are the quantifiers arising from subobjects.] this gives the definition of a *tripos* (see Pitts [1981] and Hyland, Johnstone & Pitts [1980], which perform the above construction in detail). For each object $A \in \mathcal{B}$ there is a *powerobject* $PA \in \mathcal{B}$ and a *membership predicate* $(\in_A) \in \mathtt{P}(A \times PA)$ such that given any $\phi \in \mathtt{P}(A \times B)$ we have $\ulcorner\phi\urcorner : B \to PA$ in $\mathcal{B}$ with $\mathtt{P}(1_A \times \ulcorner\phi\urcorner) \in_A \cong \phi$.

A tripos provides all of higher-order logic with the exception of *equality*. The above result essentially amounts to adding equality to the theory, and in the case of a tripos we thereby obtain a topos. This is not, however, necessarily Grothendieck. This construction was abstracted from that of the Effective topos, which is obtained from a tripos in which the predicates over X are $\mathbb{N}$-indexed sequences (R_n) of subsets of X and $(R_n) \vdash (S_n)$ if there is a recursive function $\phi : \mathbb{N} \to \mathbb{N}$ such that $\forall n \in \mathbb{N}. R_n \subset S_{\phi(n)}$.

§**3.2.19** In this section we have been describing the use of indexed category theory for many purposes in algebra and logic. One should not, however, be fooled by this into believing that it is the only or even necessarily the best tool afforded by (contemporary) category theory. I like to think of category theory as like the children's building kit "LEGO": it doesnt come with an instruction book and only lays down very primitive rules as to what you have to do with it.

In many cases we may have a generic object in the fibres, making them look like the "hom-sets" of the base category. When the fibres themselves have additional structure, this is inherited by the base category, and we say that it is *enriched* over the category describing that structure. See Kelly [1982]. Frequently this will be **Cat** itself, so that between the morphisms ("functors" or "paths") in the fibres there are 2-morphisms ("natural transformations" or "homotopies"). In the case of categories of domains we always have enrichment over **IPO**. Enriched or 2-categories my be quite capable of carrying the structure we have in mind.

§**3.2.20** The earliest example of enrichment over anything other than **Set** is the case of **Rng** over **AbGp**. In this case the rôle of the product is played by the *tensor product*. We may use this example to show that proliferated substitution is strictly more powerful than simultaneous substitution.

In discussing substitution in §§3.2.3-5 and the λ-calculus in §3.2.12, we could have allowed $[\vec{x}\,;\vec{A}]$, the collection of things of type(s) $\vec{A}$ in free variables $\vec{x}$, to be an object of some category $\mathcal{U}$ other than **Set**. Suppose this carries a tensor product, *i.e.* a functor $\otimes : \mathcal{U}\times\mathcal{U}\to\mathcal{U}$, which is associative up to coherent natural isomorphism and has a unit object $I\in\mathcal{U}$. Then substitution naturally gives rise to a $\mathcal{U}$*-enriched category*, the product being replaced by tensor product.

$$I \xrightarrow{\ulcorner x \urcorner} [x\,;X]$$

$$[x\,;Y]\otimes[y\,;Z] \xrightarrow{\ulcorner y :=_{Y;Z} \urcorner} [x\,;Z]$$

The product corresponding to concatenation of strings of terms, however, appears still to be the ordinary categorical product. We can see this in the category of sets and partial functions, where the tensor product $A\otimes B$ is the cartesian product of sets and the categorical product is $A\times B\cong A\otimes B\cup A\cup B$. This requires substitution to be strict whilst allowing pairs to be defined without both components being so.

If we attempt to replace partial functions by abelian groups with the traditional tensor product we find that simultaneous substitution is simply matrix multiplication, but that proliferated substitution is nonlinear.

§**3.2.21** Finally we remark that we may have *accidental* structure. The best known example of this is the centre of a group: all groups have centres, but the centre is not preserved by group homomorphisms. There are many more examples of this in topos theory, and the structure of an (elementary) topos is a prime example: every topos has a subobject classifier Ω (and higher order logic resulting from it) but this is not preserved by geometric morphisms. For this reason there are two sides to the logical interpretation of toposes: we may use their rich structure to prove (higher order) properties inside them, but unless our constructions are geometric they will not be preserved.

3.3 Generic Constructions

§**3.3.1** In this section for the first time we discuss type polymorphism, in the weak case of constructions which are *uniform* in some variable type. We use the determinant example introduced in section 1, but ignore the extra degree of polymorphism over the size of the matrix: the calculation of the formula $a\times d-b\times c$ suffices to illustrate the point.

A typical polymorphic language might express the 2×2 determinant function as follows:

$$\mathbf{let}\ det2((R,0,1,+,-,\times):ring\,;\ \ a,b,c,d:R)\quad=\quad a\times d-b\times c$$

where the specification of the higher type *ring* requires the provision of elements $0,1:R$ and functions $+,\times : R\times R\to R$ and $-:R\to R$.

We leave it to the designer of the language and its compiler to decide how to deal with the overloading of the symbols 0, 1, $+$, $-$ and $\times$, and concentrate on the natural categorical semantics of such a program. Since our concern is the meaning of the "compound" variable $(R,0,1,+,-,\times)$ rather than recursion, we shall here treat all functions as total.

§**3.3.2** What is the "type" of the compound variable $(R,...)$, which is to say, over what does it range? Clearly over all (commutative) rings, and we need to specify explicitly the operations in order that the compiler may arrange for calls to the relevant subroutines may be put in the code.

However although the variable itself can only denote a ring rather than a homomorphism, this higher type really is the *category* rather than the *class* of all rings. To see this we need to consider the nature of the uniformity of the definition.

Of course R, as opposed to $(R, 0, 1, +, -, \times)$, denotes the underlying set (or type) of ring elements. Admitting for the moment the claim that the *category* of rings arises, this is the *underlying set functor* $R : \mathbf{Rng} \to \mathbf{Set}$.

What then are 0, 1, $+$, $-$ and $\times$? First we have to say what $R \times R$ is. Of course it is the functor, again $\mathbf{Rng} \to \mathbf{Set}$, assigning to a ring the cartesian square of its underlying set. Then $+$ and $\times$ are natural transformations $R \times R \to R$, whilst $- : R \to R$ and $\ulcorner 0 \urcorner, \ulcorner 1 \urcorner : 1 \to R$ where $\ulcorner 1 \urcorner$ is the functor constantly assigning the singleton set.

These things *are* natural transformations because if we "replace" R by S "along" the homomorphism $\theta : R \to S$ then the operations are translated accordingly. Indeed this is precisely the definition of a *homomorphism* in the first place: we have merely transformed it to make $+$ and $\times$ the subject.

$$\begin{tikzcd} 1 \arrow[r, equal] \arrow[d, "\ulcorner 0_R \urcorner"'] & 1 \arrow[d, "\ulcorner 0_S \urcorner"] \\ R \arrow[r, "\theta"] & S \end{tikzcd} \qquad \begin{tikzcd} R \arrow[r, "\theta"] \arrow[d, "-_R"'] & S \arrow[d, "-_S"] \\ R \arrow[r, "\theta"] & S \end{tikzcd} \qquad \begin{tikzcd} R \times R \arrow[r, "\theta \times \theta"] \arrow[d, "+_R"'] & S \times S \arrow[d, "+_S"] \\ R \arrow[r, "\theta"] & S \end{tikzcd}$$

and likewise for 1 and $\times$.

§**3.3.3** We can say a little more about R and $R \times R$ than merely that they are functors. They preserve all *limits*, indeed they have left adjoints: the left adjoint of R is of course the *free ring* functor. However of more immediate interest to us now is that they preserve *filtered colimits*; this is precisely because the theory of rings is *finitary*. Since this is the generalisation of preserving directed sups, we say these functors are *continuous* as as in §2.2.7. This leads naturally to the category of all Scott continuous functors $\mathbf{Rng} \to \mathbf{Set}$ and natural transformations between them.

There is a simplification to be made here. $\mathbf{Rng}$ is an example of a *locally finitely presentable* category, which is the direct analogue of an algebraic lattice. There is a subcategory $\mathbf{Rng}_{fp}$ of finitely presentable objects (which have no proper expression as a filtered colimit) of which any object can be expressed as a filtered colimit. This means that $\mathbf{Rng} \simeq \mathsf{Ind}\,\mathbf{Rng}_{fp}$, so there is a natural equivalence between *continuous* functors $\mathbf{Rng} \to \mathbf{Set}$ and *all* functors $\mathbf{Rng}_{fp} \to \mathbf{Set}$, just as any continuous function from an algebraic lattice is determined by its effect on the compact elements, and any monotone function on them will do.

§**3.3.4** The functor category $[\mathbf{Rng}_{fp}, \mathbf{Set}]$ inherits a great deal of structure from $\mathbf{Set}$, and in fact it is a topos. On the other hand, as we shall see, it also encodes the "generic" structure of rings, and for this reason we shall call it more briefly $\mathbf{Set}[R]$.

The objects of $\mathbf{Set}[R]$ should be thought of as sets dependent upon one variable R of type ring. To start with, $\mathbf{Set}[R]$ includes all the "constant" sets (constant functors) independent of the variable R. But also the forgetful functor $R : \mathbf{Rng}_{fp} \to \mathbf{Set}$ is an object of $\mathbf{Set}[R]$, and $R \times R$ is its product with itself.

We may perform lots of other operations on R as if it were a set. In particular since $\mathbf{Set}[R]$ is a topos we may find its powerset, Ω^R. However this is an example of the *accidental* structure we mentioned in §3.2.21, because it turns out not to be preserved. We shall return to the question of preservation in §3.3.16.

In fact only those constructions involving finite limits and arbitrary colimits are preserved by the transformations we have in mind, so only these are of interest in this context. Regarding limits and colimits as "multiplication" and "addition" of sets, $\mathbf{Set}[R]$ is the collection of "polynomials" in an indeterminate R.

Examples of such structures include the circle $S = \{x, y : x^2 + y^2 = 1\}$ and the general linear group $GL_2 = \{a, b, c, d : \exists u. u(ad - bc) = 1\}$. The first is carved out as an equaliser (of the interpretations of the sides of the equation as functions $R^2 \to R$) and the second is the image of a subobject of R^5 under the product projection $R^5 \to R^4$.

§**3.3.5** In fact what we have discovered is the language of *geometric logic*. In terms of the motivations of section 2, the operations in our language are *finite limits* and *arbitrary colimits*, such that finite limits commute with *filtered* colimits. Lawvere's presentation of a finitary theory as a lex category in §3.2.8 is the algebra of "polynomials" in the theory together with finite limits, and we found that interpretations and models were given in terms of the relevant homomorphisms, which were lex functors. $\mathbf{Set}[R]$ is the algebra whose operations now include colimits, and the relevant homomorphisms preserve these too. Moreover we can present these algebras in terms of generators and relations: this is precisely how a topos is given by a *site*.

Unfortunately it is not easy to define a topos to be a category with finite limits distributing over arbitrary (small) colimits because of the usual size problems, even though this is the "algebraic" structure which we wish it to have for the purpose of defining homomorphisms (inverse image functors).

This problem is circumvented by clever use of the adjoint functor theorem. We formulate the existence of all colimits in the topos by use of indexed category theory (§4.1.10) and their preservation by inverse image functors by requiring the latter to have right adjoints. However it is the "higher order" part of the definition of an *elementary* topos (§4.4.6) which ensures that the adjoint functor theorem may be applied. The *Giraud theorem* (see, *e.g.*, Johnstone [1977], theorems 0.45 and 4.41) shows that this gives what we want.

§**3.3.6** The classifying topos is the uppermost stage in the process of "completing" a theory under all kinds of structure. The most basic example of this it the consideration of a *group* as a totality rather than as a "system of substitutions" generated by some given ones. In §3.2.7 we took an algebraic theory presented as a *signature* and built from it its *clone* and then its *Lawvere presentation*, and in the following paragraph added all finite limits to this. The classifying topos is the result of adding arbitrary colimits to this structure; this admits the facility of prescribing relations for the colimits, which is precisely the function of a Grothendieck topology (see §§3.3.11-13).

For a finitary algebraic theory there is a simple connection between the lex presentation, $\mathcal{B}$, the classifying topos $\mathcal{E}$, the category of models, $\mathcal{C}$. We obtain $\mathcal{E}$ by freely adjoining arbitrary (small) colimits to $\mathcal{B}$ and $\mathcal{C}$ by adjoining filtered colimits to $\mathcal{B}^{op}$. In symbols, $\mathcal{E} \simeq \mathbf{Set}^{\mathcal{B}^{op}}$ and $\mathcal{C} \simeq \mathsf{Ind}\,\mathcal{B}^{op}$. Conversely $\mathcal{B} \simeq \mathcal{C}^{op}_{fp}$.

§**3.3.7** The natural transformations $0, 1, +, -$ and $\times$ provide R with a ring structure in $\mathbf{Set}[R]$ and $det2$ is the 2×2 determinant map on R. We call $(\mathbf{Set}[R], R)$ the *generic ring* for the following reason.

Lemma Let A be any ring. There is a geometric morphism $\ulcorner A \urcorner : \mathbf{Set} \to \mathbf{Set}[R]$ whose inverse image functor $\ulcorner A \urcorner^*$ takes R to A, and this is unique up to isomorphism. Indeed this defines an equivalence between the category of rings and the category of geometric morphisms $\mathbf{Set} \to \mathbf{Set}[R]$.

Proof Let $X \in \mathbf{Set}[R]$, *i.e.* $X : \mathbf{Rng}_{fp} \to \mathbf{Set}$; then as remarked in §3.3.3, X extends (uniquely up to isomorphism) to a continuous functor $\mathbf{Rng} \to \mathbf{Set}$. $\ulcorner A \urcorner^*(X)$ is given simply as the value of this functor at A. It is easy to verify that $\ulcorner A \urcorner^*$ is a functor, it sends R to A and will be unique. It preserves finite limits because in $\mathbf{Set}$ (and in any finitary algebraic category over a topos) these commute with filtered colimits.

For $Y \in S$, $\ulcorner A \urcorner_*(Y) : \mathbf{Rng}_{fp} \to \mathbf{Set}$ takes B to $Y^{\mathbf{Rng}(B,A)}$. Again we have to verify that this is right adjoint.

This construction is functorial in A. Conversely the inverse image functor of any geometric morphism $f : \mathbf{Set} \to \mathbf{Set}[R]$ picks out a ring $f^*(R)$ in $\mathbf{Set}$, and we verify that this is an equivalence. $\square$

Definition Let $\mathcal{T}$ be a geometric theory. A topos, denoted by $\mathbf{Set}[T]$, is a *classifying topos* for $\mathcal{T}$ if for any topos $\mathcal{E}$ over $\mathbf{Set}$, the category of models for $\mathcal{T}$ in $\mathcal{E}$ is naturally equivalent to the category of geometric morphisms from $\mathcal{E}$ to $\mathbf{Set}[T]$.

It is a fact that any geometric theory has a classifying topos (unique up to equivalence), and every Grothendieck topos arises in this way. We shall sketch the construction of the classifying topos for a *coherent* theory.

§**3.3.8** Our polymorphic 2×2 determinant map, $det2$, is now the *generic* one in the sense that the determinant for any ring in any topos is given as the image of the generic one (§3.2.5) under an inverse-image functor of a geometric morphism.

I claim that this is the appropriate understanding of the sense in which $det2$ is polymorphic.

§**3.3.9** $\mathbf{Set}[R]$ is the *space* of all rings in a certain sense. It is a category, as is the category of rings, but they are not equivalent: the sense is a different one from this.

Given any topological space X, we may construct the category $\mathbf{Shv}(X)$ of *sheaves* on it. There are many standard works (*e.g.* [Tennison 1975]) on this, so I shall not do it here. $\mathbf{Shv}(X)$ is a topos. Given a continuous function $f : X \to Y$ between spaces, there is a geometric morphism $\mathbf{Shv}(X) \to \mathbf{Shv}(Y)$ between the toposes. Moreover if we restrict to *sober* spaces (or alternatively move to *locales*), every geometric morphism arises from a unique continuous function; in order words $\mathbf{Shv} : \mathbf{Loc} \to \mathbf{Top}$ is a full embedding.

According to Grothendieck, this means that a topos is a generalised space and a geometric morphism is a continuous function between spaces. The topos $\mathbf{Set}$ corresponds to the singleton space and so geometric morphisms $\mathbf{Set} \to \mathcal{E}$ are "points" of $\mathcal{E}$.

The points of the classifying topos $\mathbf{Set}[R]$ correspond to rings, and the natural transformations between geometric morphisms give homomorphisms of rings.

§**3.3.10** We may consider just the topos $\mathbf{Set}[R]$ without mentioning its generic ring R. The geometric morphism $\ulcorner A \urcorner : \mathbf{Set} \to \mathbf{Set}[R]$ now no longer has a particular aspect of structure to pick out, and so we must think of it as yielding the totality of all definable structure arising from the ring A. Thus there is for instance the set of quadruples (a, b, c, d) from A satisfying $(\exists u)[u(ab - cd) = 1]$, *i.e.* the set of invertible 2×2 matrices over A.

Of course in practice when we consider a ring we are interested not just in the primitive structure but also in more complex structures defined from it. This also means that we get the same answer if we vary the presentation of the theory.

A famous example of this is the equivalence between Boolean algebras (defined in terms of *true*, *false*, *and*, *or* and *not*) and Boolean rings (in which every element is idempotent). In this case the classifying toposes coincide (are equivalent) and the two (generic structures on the) generic objects are mutually definable in the obvious way.

§**3.3.11** This kind of construction may be performed for any *geometric theory*. A geometric language, like a first order language, has type, function and relation symbols of various signatures, and terms are built up from these. Geometric formulae are built up from relations (including equality) between terms together with finite conjunction, *arbitrary* disjunction and existential quantification. In coherent formulae we allow only finite disjunction.

We have essentially already described the interpretation of a geometric formula. Relations are interpreted as given subobjects, and in particular the solution-set of an equation is the equaliser of the interpretations of the terms; these are called *atomic formulae*. Conjunction and disjunction are interpreted by intersection and union of subobjects, and existential quantification by image factorisation. All of these constructions are preserved by inverse image functors.

A geometric (coherent) theory in a language is given by a collection of axioms or sequents of the form $(\phi \vdash \psi)$, where ϕ and ψ are geometric (coherent) formulae. A structure for the language *satisfies* the sequent if $[\![\phi]\!] \subset [\![\psi]\!]$ as subobjects of $X_1 \times ... \times X_n$ where X_1, ..., X_n are the types of the free variables x_1, ..., x_n of the formulae. It's easy to see that any collection of sequents is equivalent (so far as satisfaction is concerned) to one in which the antecedents are (possibly empty) conjunctions of atomic formulae and the consequents are disjunctions of formulae of the form $\exists y_1 ... \exists y_m . \psi_1(\vec{x}, \vec{y}) \wedge ... \wedge \psi_r(\vec{x}, \vec{y})$.

§**3.3.12** A geometric sequent is called *essentially algebraic* if the consequent involves no disjunction, and only *provably unique* existential quantification. It is appropriate to separate the essentially algebraic sequents in the axiomatisation of a geometric theory $\mathcal{T}$ from the other sequents. This gives a finitary many sorted essentially algebraic theory, $\mathcal{T}_0$, which can be described by a lex category $\mathcal{B}$ in the manner of Lawvere. The models in $\mathcal{E}$ of this subtheory form a category $\mathsf{Mod}_{\mathcal{E}}(\mathcal{T}_0) \simeq \mathsf{Ind}\, \mathcal{B}^{op}$, and its classifying topos is $\mathbf{Set}[T_0] \simeq \mathbf{Set}^{\mathcal{B}^{op}}$.

The models of $\mathcal{T}$ in $\mathcal{E}$ form a full subcategory $\mathsf{Mod}_{\mathcal{E}}(\mathcal{T})$ of $\mathsf{Mod}_{\mathcal{E}}(\mathcal{T}_0)$, consisting of those structures satisfying the sequents. Regarding these as "points" of classifying toposes, it is natural to think of the classifying topos $\mathbf{Set}[T]$ for $\mathcal{T}$ as a "subtopos" of $\mathbf{Set}[T_0]$, and indeed it is. It is constructed by means of a site; the category is $\mathcal{B}$, $\mathbf{Set}[T_0]$ is the category of presheaves and $\mathbf{Set}[T]$ the category of sheaves for a certain Grothendieck topology manufactured out of the axiomatisation.

§**3.3.13** An algebraic formula ϕ (finite conjunction of atomic formulae in variables $x_1, ..., x_n$) describes a certain particular finitely presented structure A of signature T_0, namely the one generated by $x_1, ..., x_n$ and satisfying the relations and equations making up the formula. The interpretation $[\![\phi]\!]$ of the formula is the set of images of this structure in the model.

Now consider a sequent $(\phi \vdash \psi)$ of the particular form in which ψ already implies ϕ and does not involve disjunction (we replace ψ by $\psi \wedge \phi$ to achieve the first, and later consider a disjunction of such sequents to accommodate the second). ψ is another algebraic formula (corresponding to a structure B), existentially quantified over the extra variables $y_1, ..., y_m$. This sequent may therefore be described by (and henceforward written as) a homomorphism $A \to B$. The model X satisfies this sequent iff any homomorphism $A \to X$ can be extended (not uniquely) to one $B \to X$ making the triangle commute.

$$\begin{tikzcd} A \arrow[r, "\phi \vdash \psi"] \arrow[d] & B \arrow[dl, dotted] \\ X & \end{tikzcd}$$

A model X satisfies a disjunction of sequents, $\phi \vdash \bigvee \psi_i$ with the same antecedent iff any homomorphism $A \to X$ may be extended to one $B \to X$ for one of the sequents or homomorphisms $A \to B$. Let R be the set of these for this disjunction. W.l.o.g. we may enlarge R to include any $A \to B'$ where $(A \to B) \in R$ and $B \to B'$ is arbitrary; for on the one hand we only need have *some* member of R, so it does no harm to increase the choice, and conversely if we manage to extend $A \to X$ to $B' \to X$ then we have $B \to B' \to X$.

A collection of homomorphisms R of this kind is called a *sieve* or *crible* on A. When we embed the opposite of the category in the classifying topos by means of the Yoneda embedding, sieves on representable objects correspond to their subobjects. A sieve which arises from the theory as above is said to be *covering*; it says that this subobject, though not necessarily presented as such, is intended to be the whole of the object.

The axiomatisation of the theory now specifies a collection $J(A)$ of covering sieves for each object A of the original category (we have one R for each time A occurs as an antecedent, so this is the infinite *con*junction). J is called a *Grothendieck cotopology*.

By now the reader will be familiar with the necessity of ensuring that concepts be preserved by substitution. In this case this condition manifests itself as follows. If $R \in J(A)$ and $\theta : A \to A'$ is any homomorphism, then $\theta^* R$, which consists of those $A \to B$ in R which factor through $A \to A'$, is to be in $J(A')$.

Using the cotopology we may now extract from amongst the presheaves $\mathbf{Set}[T_0]$, the sheaves which form the subtopos $\mathbf{Set}[T]$. Specifically, a sheaf is a presheaf which, as a functor, takes covers to colimits. Thus the purpose of a topology is to *force* the colimit structure of a category. This gives the classifying topos for $\mathcal{T}$.

[It's not clear whether the presentation of this in terms of $\mathcal{C}$ or $\mathcal{C}^{op}$ is clearer, hence the *co*'s in the words (some of which have been omitted anyway). This is because of the duality in proposition 3.2.4.]

§**3.3.14** The simplest nontrivial geometric theory is that of an "object" (with one type and no functions, relations or axioms). The classifying topos for this is written $\mathbf{Set}[U]$. This performs a similar rôle in **Top** to that performed by the Sierpiński space 2 in **Sp**. Recall from §2.3.2 that the latter *classifies open sets* and is a cogenerator for **Sp** (or **Loc**); likewise $\mathbf{Set}[U]$ cogenerates **Top**.

Since **Set** is really the one point space, it is the terminal object of **Top**, *i.e.* there is a unique (up to isomorphism) geometric morphism to it from any given (Grothendieck) topos $\mathcal{E}$. The inverse image functor of this is written Δ, and is the "constant object" functor (*cf.* the combinator K) if we think of $\mathcal{E}$ as a category of functors from a lex category $\mathcal{C}$ into **Set**. The direct image, traditionally called Γ, is the "global section" functor and is given by evaluation at 1 (the terminal object of $\mathcal{C}$).

In the case $\mathcal{E} = \mathbf{Set}[U]$, Δ also has a left adjoint Σ given by evaluation at 0 (the empty finite set). A geometric morphism f for which f^* has a left adjoint $f_!$ as well as a right adjoint f_* is called *essential*. [In fact the functor $\Sigma : \mathbf{Set}[U] \to \mathbf{Set}$ in turn has a left adjoint; this is called π_0 because in topology it is associated with the set of components of a space.]

§**3.3.15** This provides us with a simple interpretation for type polymorphism of sets (readily extendable to include models of any geometric theory). Recall the view that toposes are the algebras for the language of geometric logic, and index them over the "free variables"; the substitution homomorphisms are inverse image functors of geometric morphisms.

What are these "free variables"? Of course they are like our $(R, 0, 1, +, -, *)$, generic models of geometric theories. The base (syntactic) category is just the category of geometric theories and their interpretations.

Let $\mathcal{T}_1, \mathcal{T}_2$,... be geometric theories and $\mathbf{Set}[T_1]$, $\mathbf{Set}[T_2]$, ... their classifying toposes. From these it is possible to construct $S[T_1, T_2, ...]$, the simultaneous classifying topos for models of each of the theories, as a pullback in **Top**. This contains generic models $T_1, T_2, ...$ for $\mathcal{T}_1, \mathcal{T}_2, ...$ respectively and is the category of all structures definable in terms of them. We may therefore consider it the category of type expressions in the free type variables $T_1, T_2, ...$ of appropriate kinds.

Now in this case we have more than we did with ordinary algebraic theories, because the fibres are categories rather than discrete sets and the substitution maps are functors. Hence we have an indexed type theory and we can ask about sums, products and exponentials as in §4.1.

§**3.3.16** Since the substitution functors are inverse images, they preserve geometric logic. However they do not preserve the other logical structure which we find *inside* a topos — it is *accidental* in the sense of §3.2.21. A functor between toposes which preserves finite products, exponentials and the subobject classifier is called a *logical functor* because it preserves all higher order logic; we shall discuss weaker things than this.

In order to obtain the structure of a hyperdoctrine we need a left as well as a right adjoint to the inverse image functor (giving an *essential* geometric morphism) and the Beck condition. Such a geometric morphism is said to be *locally connected* [Johnstone 1980]; the weaker poset form of this condition of being *open* is equivalent to preserving first order logic.

In fact we do not need the *left* adjoint to ask about the Beck condition. Suppose we have a pullback of geometric morphisms in **Top** as follows:

$$\begin{tikzcd}
\mathcal{F}\times_{\mathcal{E}}\mathcal{G} \arrow[r, "h"] \arrow[d, "k"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{G} \arrow[d, "g"] \\
\mathcal{F} \arrow[r, "f"] & \mathcal{E}
\end{tikzcd}$$

Then there are two forms of the condition; we say that f *satisfies the Beck condition on the left* if, for all g with the same codomain, the first diagram of functors below (corresponding to the above pullback) commutes:

$$\begin{tikzcd}
\mathcal{F}\times_{\mathcal{E}}\mathcal{G} \arrow[r, "h_*"] & \mathcal{G} \\
\mathcal{F} \arrow[u, "k^*"] \arrow[r, "f_*"] & \mathcal{E} \arrow[u, "g^*"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{F}\times_{\mathcal{E}}\mathcal{G} \arrow[d, "k_*"'] & \mathcal{G} \arrow[l, "h^*"'] \arrow[d, "g_*"] \\
\mathcal{F} & \mathcal{E} \arrow[l, "f^*"']
\end{tikzcd}$$

One may show that this is equivalent to local connectedness. Likewise we say that f *satisfies the Beck condition on the right* if we have this for the second diagram. Lindgren [1984] has shown that this holds iff f_* preserves filtered colimits.

3.4 Polymorphic Lambda Calculus

§**3.4.1** So far, all the forms of polymorphism we have discussed have been of a "functorial" form interpretable in **Set**, or at least in some topos. The ideas which form the subject of the remaining two sections of this chapter are from a different tradition and turn out to be incompatible with the "ordinary" logic of sets; indeed there are several paradoxes to this effect, to which we shall add a new one in theorem 5.5.9.

The underlying idea, which first occurred in Mathematical Logic though the foregoing discussion emphasises its relevance to Computer Science, is that the identity function, $\lambda x.x$, for instance, does not occur separately as 1_X once for each type X, but in a single "Platonic" incarnation of "type" $\forall X.X \to X$, which is instantiated for each type X in a fashion similar to the use of the quantifier in predicate logic. The aim of this section is to provide a formal justification for the use of the quantification symbol in the shape of a demonstration that it is right adjoint to substitution.

There is a remarkable by-product of this approach. Church's numerals (§1.1.7) are of type $\forall X.(X \to X) \to (X \to X)$, and in certain interpretations it may be shown that these are the only terms of this type. Accordingly we have an object which looks like the natural numbers. Likewise the object $\forall U.(X \to U) \to ((Y \to U) \to U)$ is similar to a sum, despite the paradox corollary 1.5.12.

§**3.4.2** We look first at a second-order polymorphic lambda calculus studied by Bruce and Meyer [1984]. This admits type variables and expressions with quantification, but allowing only "sets" and not "structures".

It has

(i) 0-variables x, y, z, ...

(ii) 1-variables X, Y, Z, ...

(iii) (0-)terms formed by the clauses

$$x \qquad (\lambda x : B.a) \qquad (ab) \qquad (\Lambda X.a) \qquad a^A$$

(iv) types (1-terms) formed by the clauses

$$X \qquad (A \to B) \qquad (\forall X.A)$$

(v) α_0, β_0 and η_0 rules

$$\begin{array}{lcl} \lambda y : B.a & = & \lambda x : B.a[y := x] \\ (\lambda x : B.a)b & = & a[x := b] \\ \lambda x : B.ax & = & a \end{array}$$

(where in the first and third clauses x is not free in a),

(vi) α_1, β_1 and η_1 rules

$$\begin{array}{lcl} \forall Y.A & = & \forall X.A[Y := X] \\ \Lambda Y.a & = & \Lambda X.a[Y := X] \\ (\Lambda X.a)^B & = & a[X := B] \\ \Lambda Y.a^Y & = & a \end{array}$$

(where in the first clause X is not free in A, and in the second and fourth it is not free in the type of any free variable of a),

(vii) non-unique typing of terms

$$\begin{array}{llll} & & & x : A \\ \text{if} & f : (A \to B) \text{ and } a : A & \text{then} & (fa) : B \\ \text{if} & a : A & \text{then} & (\lambda x : B.a) : (B \to A) \\ \text{if} & a : A & \text{then} & (\Lambda X.a) : (\forall X.A) \\ \text{if} & a : (\forall X.A) & \text{then} & a^B : A[X := B] \end{array}$$

(viii) equality rules on 0-terms as appropriate.

By convention $\forall$ governs as much of the expression as possible to the right, so $\forall X.X \to X$ means $\forall X.(X \to X)$ and not $(\forall X.X) \to X$.

+§**3.4.3** Now we shall present this as an indexed category in a manner generalising §1.3.10. The objects of the base category will count the free 1-variables in the terms of the corresponding fibres — we shall set aside the additional complication of counting 0-variables as in §3.2.2.

Let the base category have *objects* the "products" (*i.e.* strings) of 1-variables and *morphisms* from $\vec{Y}$ to $\vec{X}$ the strings of length $|\vec{Y}|$ of types with free variables among $\vec{X}$. Variables provide the *identity* and substitution the *composition*. *Product* is given by concatenation. Hence we have a category with products.

The fibre $\mathsf{P}\vec{X}$ over the string $\vec{X}$ has *objects* the products (strings) of types $\vec{A}$ with free 1-variables among $\vec{X}$ and the *substitution* functor over $\vec{B} : \vec{Y} \to \vec{X}$ acts on objects as the substitution $[\vec{X} := \vec{B}]$ of free type variables. Accordingly the object parts of the fibres are the regular representation of the base category (§3.2.4).

The *morphisms* of the fibre $\mathsf{P}\vec{X}$ from $\vec{B}$ to $\vec{A}$ are the strings of terms of types $\vec{A}$ with free 0-variables of types $\vec{B}$. As usual *identity* and *composition* arise from variables and substitution. The effect of the *substitution* functor is again the substitution $[\vec{X} := \vec{C}]$. The fibres have *products* by concatenation.

The fibres also have *exponentials* in the same way as the λ-calculus with fixed types. This arises from the constructor $(A \to B)$ with the obvious iteration to provide from strings. We need the η_0-rule to ensure that the adjunction is bijective. Again exponentials are preserved by substitution.

+§**3.4.4** The type variable X in the fibre over X is a *generic type* in the sense of §3.2.5. So for any type A in the fibre over $\vec{Y}$ there is a unique morphism, called A, from $\vec{Y}$ to X the effect of whose substitution at X over X is A over $\vec{Y}$.

+§**3.4.5** The significant point of this language and indexed presentation is that $\forall$ and Λ provide quantification over product projections in the base category, *i.e.* right adjoints to the corresponding substitution functors.

For we have a natural bijection

$$\frac{\vec{B} \longrightarrow \vec{A}}{\vec{B} \longrightarrow \forall X.\vec{A}}$$

where X is not free in $\vec{B}$. For above the line we have the terms $\vec{a}$ of type $\vec{A}[X]$ in free variables of types $\vec{B}$, and these correspond to terms $\vec{a}' = \Lambda X.\vec{a}$ of type $\forall X.\vec{A}$ in the same free variables since $\vec{a} = (\Lambda X.\vec{a})^X$ and $\vec{a}' = \Lambda X.(\vec{a}')^X$ by the β_1 and η_1 rules.

It is an easy matter to verify that the Beck condition holds.

The syntax provides no means to express quantification over other maps in the base category, as we may see from their reading in Natural Deduction as in §3.2.16.

§**3.4.6** Seely [1986] discusses a language due to Girard [1972] which is "higher-order" in the sense that it admits variables for type-valued functions of types.

The difference in the interpretation is that now we require the base category to be cartesian closed.

§**3.4.7** We are now in a position to propose a provisional definition for a model of a language of this kind. It is an indexed category of a certain form. The base category has at least products and desirably exponentials; it may as well be generated by an object V which is a metatype of types. The fibre over a power of V, which is the term structure with this many free type variables, is to be cartesian closed, and this structure is to be preserved by substitution. The types (objects of the fibre) are *not* to be functorial in the type variable, but the substitution *is* to be a functor. Finally we require substitution over at least the product projections to have a right adjoint; in §4.1.7 we shall see why we call this "completeness" for the category.

A presentation of this might be as follows [Moggi 1986]. The types form a small (internal) category (§3.2.8, 4.2.10, 5.5.2) $\mathcal{C}$ with object set C_0. A type A with free variables $\vec{X}$ is interpreted as *any* function $C_0^n \to C_0$, where $n = |\vec{X}|$. This is *not* to be a functor.

A term a of type A with free variables of types $\vec{B}$, where A and $\vec{B}$ themselves have free type variables $\vec{X}$, is a function ${C_0}^n \to C_1$ (where $n = |\vec{X}|$) whose postcomposites with the domain and codomain maps $C_1 \rightrightarrows C_0$ are the interpretations of $\vec{B}$ and A respectively. This interpretation is generated in the obvious way from variables as product projections and application and abstraction as the exponential adjunction.

In order to interpret the quantified type $\forall X.A$ we need a right adjoint Π to the substitution (inclusion) $C_0 \to {C_0}^{C_0}$. We then proceed as in §3.4.5.

The problem with this approach is Freyd's paradox (proposition 1.5.8). This simply means that $\mathcal{C}$ cannot be an internal category in **Set** since that argument depended upon excluded middle. We shall show that it *is* possible to do this kind of thing with domains instead of sets. It is also possible to do it in the Effective Topos [Hyland 1982].

§**3.4.8** From languages of purely theoretical interest we shall now turn to one called PONDER which has been implemented practically by Fairbairn [1984]. His emphasis is upon the use of types as a programming aid (§3.1.1) to ensure that functions are only applied to arguments of appropriate types, and so this work concerns a *type-checking algorithm*. Instantiation for X in $\forall X.A$ is then a form of *coercion* (§1.3.8) and so we have to consider the rules for this.

The syntax of PONDER is essentially as in §3.4.2, with the additional features of *casts* (expressions declared, perhaps erroneously, to be of a certain type), *capsules* (declared types which must match by name and not structure), *overloaded* operators (§3.1.1) and *generators* (recursively defined types). For our purposes the first three are syntactic sugar, and we shall discuss the last in the next section.

As in §1.3.9 we have a relation $A \subset B$ between types such that whenever we have $a : A$ then we also have $a : B$. Fairbairn writes $\geq$ for this, contrary to the usual convention for the relationship between posets and categories.

§**3.4.9** This relation satisfies the following rules.

(i)				A	$\subset$	A
(ii)	if	$A \subset B$ and $B \subset C$	then	A	$\subset$	C
(iii)	if	$A \subset A'$ and $B' \subset B$	then	$(B \to A)$	$\subset$	$(B' \to A')$
(iv)				$(\forall X.A)$	$\subset$	$A[X := B]$
(v)	if	$A \subset B$ and $X \notin \mathsf{FV}(A)$	then	A	$\subset$	$(\forall X.B)$
(vi)	if	$X \notin \mathsf{FV}(A)$	then	$(\forall X.A \to B)$	$\subset$	$(A \to \forall X.B)$

We shall consider the rules for recursion in §3.5.4.

Clearly axioms (i) to (iii) are those for an (indexed) Heyting system (§1.3.9). (iv) and (v) say that $\forall X$ is right adjoint to substitution, *i.e.* a universal quantifier; the monotonicity of $\forall X$ w.r.t. $\subset$ follows from these. (vi) is in fact equivalent to preservation of $\to$ by substitution; the reverse inequality follows from the other axioms.

The essence of the idea that $\forall X.A$ is "type A, whatever the value of X", as in our "Platonic" identity of §3.4.1, is in axiom (iv).

We have now justified the claim that $\forall X$ is a right adjoint to substitution.

3.5 Strong Polymorphism

§**3.5.1** Finally we consider recursively-defined types and type-of-types, both from the historical negative point of view and in the positive case of highly-typed programming languages. Arguably the notion of a type of types has been present in Computer Science since Babbage's first *stored program*. Also, although in practice compilers are essentially syntactic beasts, in order for the compiler for a typed language itself to be well-typed, it is necessary to have a type-of-types in the language.

§**3.5.2** First we look at the recursively defined types in PONDER (§3.4.8). The syntax of this language as implemented is *declarative*, *i.e.* it allows new names to be introduced. In particular we have, for instance,

$$\begin{array}{lcl} \textbf{type}\, Identity & = & \forall X.X \to X \\ \textbf{type}\, Pair[A,B] & = & \forall X.(A \to (B \to X)) \to X \\ \textbf{rectype}\, Infinite\_List[A] & = & Pair[A, Infinite\_List[A]] \end{array}$$

Since Fairbairn's intention was to build a *terminating* type-checker, there is a restriction in the language in the form of recursive type declarations. This is that all applications of the generator (here $Infinite\_List$) within the body of the declaration must be to *exactly* the parameters as on the left hand side. This restriction ensures that such infinite behaviour as arises in the type-checker is only of a *cyclic* kind and can therefore be trapped by incorporating memory into the algorithm. Without it, type-checking would have the full power of the λ-calculus and hence be undecidable.

Our intentions, on the other hand, are semantic, and we have not worried about termination elsewhere in this work, so we shall not consider this restriction.

§**3.5.3** Here are the coercion rules for recursive types. These are more complicated than those in §3.4.9, so we have to include the assumptions (Γ) explicitly. The typing rules may be deduced from them since the intention of $A \subset B$ is that $a : A$ implies $a : B$ for all terms a.

$$\frac{\Gamma,\ A \subset B \vdash G[A] \subset B \qquad \Delta \vdash A \cong G[A]}{\Gamma \cup \Delta \vdash A \subset B}$$

$$\frac{\Gamma \vdash G[X_1,...,X_n] \cong A}{\Gamma \vdash G[X_1,...,X_n] \subset A} \qquad \frac{\Gamma \vdash G[X_1,...,X_n] \cong A}{\Gamma \vdash A \subset G[X_1,...,X_n]}$$

§**3.5.4** Since types form a category rather than a poset, the semantics of recursive types are more complicated than simple fixpoint equations. Of course we have already discussed this in §2.2.2.

§**3.5.5** Now let us turn to a "full-blown" form of type polymorphism involving type-of-types. This discussion is based on that of Cardelli [1986], which gives numerous examples of constructs which can be expressed in the language; the semantics given there is based on the $P\omega$ model (§1.4.8).

Admitting the *type* of types is the same as making types *values*, so we no longer have the stratified variables and terms which we had in the second order polymorphic lambda calculus (§3.4.2), PONDER (§3.4.8) or Martin-Löf [1975]. Types can therefore be the results and parameters of computation, in contrast to their usual passive (and solely compile-time) rôle in type-checking.

Types can also depend on non-type parameters. This is quite familiar in Mathematics, *e.g.* "the *set* of points at distance r from the origin". Notice that y is free in $\forall x : a(y).b$; again there are familiar mathematical examples.

§**3.5.6** The *terms* of Cardelli's language are as follows:

1. variables, x,
2. type of types, V,
3. abstraction, $(\lambda x : a.b)$,

4. application, (ab),
5. product, $(\forall x : a.b)$,
6. pair, $\langle a, b\rangle$,
7. component, $(a \restriction b)$,
8. sum, $(\exists x : a.b)$,
9. recursion, $(\mu x : a.b)$.

The *reduction rules* are:

$$\begin{array}{lcll}
((\lambda x : c.a)b) & = & a[x := b] & (\beta)\\
(\lambda x : c.(ax)) & = & a & (\eta)\\
\langle a_0, a_1\rangle & = & a & (\pi)\\
(\langle a, b\rangle \restriction c) & = & ((ca)b) & (\sigma)\\
(\mu x : c.a) & = & a[x := (\mu x : c.a)] & (\mu)
\end{array}$$

where in η, $x \notin \mathsf{FV}(a)$, and in π, $_0 = \lambda x : A.\lambda y : B.x$, *etc.*

§**3.5.7** Type-checking in this language may involve non-termination computation, and so is undecidable. In what follows, Γ is a *type-assignment*, *i.e.* a set of formulae $a : A$. The empty assignment is valid, and further ones are generated by the rule

$$\frac{\Gamma \text{ ok} \quad \Gamma \vdash A : \mathsf{V}}{\Gamma,\ x : A \text{ ok}} \quad x \notin \mathsf{FV}(\Gamma)$$

Now we have a long sequence of rules defining the type inference relation $\vdash$. Observe that the last of these involves computation.

$$\frac{}{\Gamma,\ x : A \vdash x : A}\,\mathrm{A} \qquad \frac{}{\Gamma \vdash \mathsf{V} : \mathsf{V}}\,\mathsf{V}\mathcal{I}$$

$$\frac{\Gamma,\ x : A \vdash B : \mathsf{V}}{\Gamma \vdash (\forall x : A.B) : \mathsf{V}}\,\forall\mathsf{f} \qquad \frac{\Gamma,\ x : A \vdash B : \mathsf{V}}{\Gamma \vdash (\exists x : A.B) : \mathsf{V}}\,\exists\mathsf{f}$$

$$\frac{\Gamma,\ x : A \vdash b : B}{\Gamma \vdash (\lambda x : A.b) : (\forall x : A.B)}\,\forall\mathcal{I} \qquad \frac{\Gamma \vdash a : A \quad \Gamma \vdash f : (\forall x : A.B)}{\Gamma \vdash (fa) : B[x := a]}\,\forall\mathcal{E}$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[x := a]}{\Gamma \vdash \langle a, b\rangle : (\exists x : A.B)}\,\exists\mathcal{I} \qquad \frac{\Gamma \vdash c : (\exists x : A.B) \quad \Gamma \vdash d : (\forall x : A.\forall y : B.C\langle x, y\rangle)}{\Gamma \vdash (c \restriction d) : Cc}\,\exists\mathcal{E}$$

$$\frac{\Gamma,\ x : A \vdash a : A}{\Gamma \vdash (\mu x : A.a) : A}\,\mu\mathsf{f} \qquad \frac{\Gamma \vdash a : A \quad \Gamma \vdash B : \mathsf{V} \quad A = B}{\Gamma \vdash a : B}\,\mathsf{Red}$$

where in $(\exists\mathcal{E})$, $x, y \notin \mathsf{FV}(C)$.

§**3.5.8** There are certain constructions which are basic to category theory but derivable from Cardelli's language (as in PONDER). These are

void (0)	$\perp$	$=$	$\lambda A : \mathsf{V}.\mu x : A.x$	$: \forall A : \mathsf{V}.A$
unit (1)	$*$	$=$	$\lambda A : \mathsf{V}.\lambda a : A.a$	$: \forall A : \mathsf{V}.\forall a : A.A$
Boolean (2)	$0, 1$			$: \forall A : \mathsf{V}.\forall a : A.\forall b : A.A$
conditional	cond	$=$	$\lambda c : 2.\lambda A : \mathsf{V}.\lambda a : A.\lambda b : A.cAab$	$: 2 \to \forall A : \mathsf{V}.A \times A \to A$
exponential	$\to$	$=$	$\lambda A : \mathsf{V}.\lambda B : \mathsf{V}.\forall x : A.B$	$: \mathsf{V} \to (\mathsf{V} \to \mathsf{V})$
product	$\times$	$=$	$\lambda A : \mathsf{V}.\lambda B : \mathsf{V}.\exists x : A.B$	$: \mathsf{V} \to (\mathsf{V} \to \mathsf{V})$
sum	$+$	$=$	$\lambda A : \mathsf{V}.\lambda B : \mathsf{V}.\forall C : \mathsf{V}.(A \to (B \to C)) \to C$	$: \mathsf{V} \to (\mathsf{V} \to \mathsf{V})$

§**3.5.9** How can we present this language as an indexed category? First we have to construct the (ordinary) category $\mathcal{C}$ of types and terms as in §1.3.10. Of course this has *objects* the strings $\vec{A}$ of types (A such that $A : \mathsf{V}$) and *morphisms* from $\vec{B}$ to $\vec{A}$ the $|\vec{A}|$-strings of types $\vec{A}$ in free variables of types $\vec{B}$. This is cartesian closed because we have an extension of the λ-calculus.

Next we have to deal with the quantifiers. This involves building an indexed category over $\mathcal{C}$. The fibre over $\vec{A}$ has *objects* the strings of type-expressions with free variables of types $\vec{A}$ and *morphisms* the appropriate strings of terms. The *substitution* functors are, of course, given by substitution.

Now how do we interpret $\forall x : B.c$, where B has free variables of types $\vec{A}$ and c has them of types $\vec{A}, B$ (the last being of course x)? The expression has free variables of types $\vec{A}$, so we have a functor from the fibre over $\vec{A}, B$ to that over $\vec{A}$; over what morphism $(\vec{A}, B) \to \vec{A}$ does this lie? This corresponds to a string of terms of types $\vec{A}$ in free variables $\vec{A}, B$; this must simply be the string of corresponding variables. However this is *not* a product projection as it was in the second order polymorphic lambda calculus or PONDER, because B itself depends on A; in fact it is the *display map* corresponding to the sum $\exists \vec{a} : \vec{A}.B$ over $\vec{A}$. We shall meet this in the case of **Set** in §4.1.11 and subsequently in other categories. $\forall x : B$ is then the right adjoint to this substitution map.

The sum $\exists x : B.c$ is similarly interpreted as the corresponding left adjoint. The Beck condition is satisfied in the usual trivial way.

The μ operator gives (or is interpreted as) indexed fixpoints, *i.e.* $\mathsf{Y}_A = \lambda f : (A \to A).\mu x : A.fx$.

$^-$§**3.5.10** Finally we have to bring V into the picture. Let X be a variable of type V. This is then an object of the fibre over X, and is such that any other type (in however many free variables of whatever type) is obtained by substitution. X is therefore generic in the sense of §3.2.5.

We are also interested in the (objects of) this fibre over X. We have a type of them, namely $\mathsf{G} = \exists X : \mathsf{V}.X \to \mathsf{V}$. We shall see in §5.4 that this is a *generic display* and is sufficient to code up the whole of the structure of the model with type-of-types.

$^-$§**3.5.11** Finally let us put this in historical perspective by surveying briefly some of the corpses in this philosphical minefield.

The first casualty was of course Frege [1893], who attempted to axiomatise Set theory (and hence Mathematics) using an *unrestricted Axiom of Comprehension* (*cf.* §1.5.11). This means that, given any formula $\phi(x)$ whatever, there is a set "$\{x : \phi(x)\}$" of all things satisfying ϕ. As is well known, Russell pointed out to the unfortunate Frege that $\phi(x) = x \notin x$ leads to a contradiction.

The main response to this has been to *bind* the set-abstraction and quantifiers. Category theorists (at least) would claim that there is no occasion in Mathematics when these operations are not implicitly bound, and in this work the common convention of using lower-case letters for variables of the corresponding upper-case type has usually been adopted.

Russell [1908] was the first to formalise this, as the *theory of types*. We build an $\mathbb{N}$-indexed hierarchy in which each level consists of all subsets of sets in the level below. This is in fact the simplification due to Ramsey: the original version took account of quantifiers as well. This was a very advanced notion for its time: the systematic classification of sets according to their quantifier complexity (most of the interest being in quantification over subsets of $\mathbb{N}$, which are usually called "real numbers") began with Kleene.

Russell made an important observation which he called *typical ambiguity*. This is that whenever we have a proof of $\phi(x)$ for x of some type n, then we also have it for x of any other type (so long as there are sufficiently many lower types for its hereditary elements). Accordingly his style is such that the variables are implicitly to be understood schematically with quantification over all types. This is of course already a form of polymorphism (we used this kind of type-shifting implicitly in §1.3.12).

Quine [1937] took this a stage further and used this polymorphism to abolish the types. Instead he stratified the formulae (by the $\in$-relation). Thus we are allowed to write $x \in y \in z$ but not $x \in x$ or $x \in y \in z ... x \in z$. Formally, there must be an assignment of natural numbers (or integers, to remove the "lower types" complication) to the variables such that $\in$ occurs only between variables of successive types. This is called *New Foundations* or *NF*. For a general review of this theory and its developments see [Forster 1983].

The remarkable thing about NF is that it allows a universal set, and also complements of arbitrary sets inside it. The Russell paradox of course cannot be stated within it. On the other hand its heavy emphasis on the $\in$ relation means that something as basic as the cartesian product construction has a very complicated expression in terms of $\langle x, y\rangle = \{\{x\}, \{x, y\}\}$.

There are subsystems of NF depending on the number of types which need to be used. The first two levels (in which $a \in b$ and $a \in b \in c$ are allowed but not $a \in b \in c \in d$) can be shown to be consistent, as can NFU (NF plus "Ur" elements or atoms). NF_4 is in fact the full theory, since it allows a kind of duality to be coded, specifically the operation $Bx = \{y : x \in y\}$.

$^-$§**3.5.12** There is another paradox, which was in fact noticed before the better known one due to Russell. This is based on the observation that ordinals (well-founded totally ordered sets) are themselves well-ordered and hence "algebraically" form an ordinal usually called **On**. Thus in the presence of unrestricted comprehension $\mathbf{On} \in \mathbf{On}$, and indeed it is the largest ordinal, contrary to the ease with which the successor may be constructed.

This, the **Burali-Forte paradox** [Rosser 1942], has perhaps been even more of a killer than Russell's. One might attempt to extend NF by allowing class variables (as in *Gödel-Bernays set theory*) and admitting as a set the intersection of any abstracted class with a set. However in this theory one may formulate the ordinal-of-ordinals and hence reach contradiction.

The most famous victim of this paradox is probably Martin-Löf's original attempt at an intuitionistic type theory [1971]. It was discovered both that it was possible to define a type of well-founded partial order types (itself a partial order) and to prove induction over well-founded partial orders. This necessitated some quite far-reaching changes in the type theory, involving the introduction of a nested sequence of universes. Martin-Löf's executioner was Jean-Yves Girard, who was also interested in this area, largely for proof theoretical reasons. He produced a system of polymorphic λ-calculus known as "Système F" [1972].

This was also a method which was suggested for dismissing typoses with equality (theorem 5.5.9), but other arguments were found instead.

Chapter 4

The Indexed Category Theory of Sets

4.1 Indexed Families, Products and Coproducts

§**4.1.1** The previous chapter set out to motivate Indexed Category Theory from a *syntactic* point of view. We aim, of course, to apply it *semantically* to categories of domains, and this we do in chapter V. First, however, we must devote the present chapter to the case of discrete sets, which we approach with a view to formulating the fact that **Set** is "complete", that it has all "small" (*i.e.* set- rather than class-indexed) limits (and colimits). The answer to this depends upon first being able to express the notion of a "family" of sets.

This chapter sets out some of the basic ideas of indexed category theory, motivated in the first instance by this problem. In the case of **Set** we would like to be able to define an indexed family of sets as a function from the indexing set to the "set" of all sets. Of course Russell showed long ago that we cannot have this. However there is a trick with disjoint unions and pullbacks which enables us to perform an equivalent construction called a *fibration*.

The opening section motivates the notion of an indexed family of sets and the definitions of indexed products and coproducts. In the second section we discuss other categories and constructions indexed over a base category. In section 3 we address the subject of cartesian closure again, this time from an *indexed* point of view. In the case of **Set** we require pullback against arbitrary maps (not just product projections) to have a right adjoint, but this is too strong a condition for categories of domains. Instead we define *relative* cartesian closure, which we find already applies to **Cat** in another illustration of our confusion of object and metalevels. In the final section we discuss the indexed forms of the general and special adjoint functor theorems and introduce elementary toposes; this illustrates the way in which adjoints can be used to replace infinitary structure such as arbitrary colimits.

§**4.1.2** We begin by formalising the notion of an indexed family of objects. This account of indexed categories is very loosely based on [Johnstone *et al.* 1978] and [Johnstone 1983].

Let $\mathcal{S}$ be some category, whose objects we are thinking of as sets — in the first instance $\mathcal{S} = \mathbf{Set}$. In order to talk about products, coproducts and so on in $\mathcal{S}$ we need some notion of an *A-indexed family of objects* of $\mathcal{S}$, for each $A \in \mathsf{ob}\,\mathcal{S}$. A naïve notation for such a thing might be $(X_a : a \in A)$. Between these there are A-indexed functions, $(f_a : X_a \to Y_a : a \in A)$, so that we have a category $\mathcal{S}^A$. In the basic example this category is simply the A-fold power of the category $\mathcal{S} = \mathbf{Set}$.

As well as A-indexed families, we have *substitution* or *relabelling* functors. If $\alpha : B \to A$ is any $\mathcal{S}$-map and $(X_a : a \in A)$ is an A-indexed family, we have a B-indexed family $(X_{\alpha b} : b \in B)$. The same applies to morphisms, so this is in fact a functor $\mathcal{S}^\alpha : \mathcal{S}^A \to \mathcal{S}^B$. The assignment $\alpha \mapsto \mathcal{S}^\alpha$ is itself (pseudo) (contra) functorial, in that $\mathcal{S}^{\mathsf{id}} \cong \mathsf{id}$ and $\mathcal{S}^{\alpha;\beta} \cong \mathcal{S}^{\beta};\mathcal{S}^{\alpha}$. These natural isomorphisms will have to satisfy some coherence conditions, but we shall not pay too much attention to them.

§**4.1.3** $(X_a : a \in A)$ appears to be a function from A to the class of all sets, which is a very troublesome notion. As has been remarked, we shall be able to think in this way in $\mathbf{Retr}(\Lambda)$, but not in **Set**. The trick in **Set** is to code this up using the disjoint union, making use of our *a priori* knowledge of the structure of the category but quietly subsuming the *Axiom of Replacement*. The indexed set $(X_a : a \in A)$ is represented by its disjoint union together with the *display map* which identifies the index:

$$\begin{tikzcd} X = \coprod_{a \in A} X_a \arrow[d] & x \in X_a \arrow[d, mapsto] \\ A & a \end{tikzcd}$$

An object of $\mathcal{S}^A$ is therefore just a function, or $\mathcal{S}$-morphism; X_a is picked out as the inverse image of a, *i.e.* the pullback of the singleton function $\ulcorner a \urcorner : 1 \to A$ against the display map. In the case $\mathcal{S} = \mathbf{Set}$ *any* map may occur as a display map.

The substitution functor $\mathcal{S}^{\alpha} : \mathcal{S}^A \to A^B$ over $\alpha : B \to A$ is easily seen to be given by pullback along α and is consequently written $\mathtt{P}\alpha$ or α^*:

$$\begin{tikzcd} \coprod_{b \in B} X_{\alpha b} \cong X \times_A B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner"] & X \cong \coprod_{a \in A} X_a \arrow[d] \\ B \arrow[r, "\alpha"] & A \end{tikzcd}$$

§**4.1.4** There is an alternative description of this set-up in terms of *fibrations*. The *objects* over $A \in \mathcal{S}$ are the $\mathcal{S}$-maps with codomain A, but besides forming categories over each A (called *fibres*) all the objects together form a category called $\mathcal{S}^2$ because it is the category of functors from $2 = (\bullet \to \bullet)$ to $\mathcal{S}$; the *morphisms* are just the commutative squares in $\mathcal{S}$. There is then a functor $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ with the property that $X \in \mathcal{S}^A$ iff $\mathsf{cod}(X) = A$ and the maps $X \to Y$ over $\alpha : B \to A$ (*i.e.* the squares whose lower side is α) correspond bijectively to the maps $X \to \mathtt{P}\alpha Y$ in $\mathcal{S}^B$. The maps within a single fibre (*i.e.* the squares with an identity along the bottom) are (for obvious reasons) called *vertical* whilst those which give pullback squares are called *horizontal* or *cartesian*.

The fibred (as opposed to indexed) approach was pioneered by Bénabou [1975] in recognition of the fact that substitution is in practice defined only up to isomorphism.

The fibre over A is in this case may be seen as either the A-fold power of $\mathcal{S}$ or as the *slice category* $\mathcal{S}/A$ whose objects are the $\mathcal{S}$-morphisms with codomain A and whose morphisms are the $\mathcal{S}$-morphisms making the triangles commute. In the relative case the objects of the slice will be display maps but the morphisms will still be arbitrary maps.

There is, however, nothing in the definition of a general fibration to require the fibre over an arbitrary A to be the A-fold power of that over the terminal object of $\mathcal{S}$. Indeed the difference is quite crucial to most applications. We shall denote the fibre over A by $\mathtt{P}A$ and the substitution functor over $\alpha : B \to A$ by $\mathtt{P}\alpha$.

Lemma Let $\alpha : B \to A$ and U, X be over B, A respectively. Then there is a natural bijection

$$\frac{U \longrightarrow_{\alpha} X}{U \longrightarrow_{B} \mathtt{P}\alpha X}$$

□

§**4.1.5** Let us consider the form of (binary) products in the fibre categories. Let X and Y be objects over A, presented either as A-indexed things or as displays ($\mathcal{S}$-morphisms with codomain A), and write $X \times_A Y$ for their product according to either interpretation. Naïvely this is $(X_a \times Y_a : a \in A)$, but as a display it turns out to be precisely the pullback, hence the alternative name *fibre product* for the latter. In English we may therefore say "fibre products are pullbacks", although the (Gaullist) French can't make the *a priori* distinction!

$$
\begin{tikzcd}
 & X \times_A Y \arrow[dl] \arrow[dr] & \\
X \arrow[dr] & & Y \arrow[dl] \\
 & A &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathtt{P}\alpha E \arrow[r] \arrow[d, hook] \arrow[dr, phantom, "\lrcorner", very near start] & E \arrow[d, hook] \\
\mathtt{P}\alpha X \arrow[r] \arrow[d, shift right, "\mathtt{P}\alpha f"'] \arrow[d, shift left, "\mathtt{P}\alpha g"] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, shift right, "f"'] \arrow[d, shift left, "g"] \\
\mathtt{P}\alpha Y \arrow[r] & Y \\
B \arrow[r, "\alpha"] & A
\end{tikzcd}
$$

The indexed equaliser of $f, g : X \rightrightarrows Y$ in $\mathcal{S}^A$ is $E = (E_a : a \in A)$ where $E_a = \{x_a \in X_a : f_a x_a = g_a x_a\}$; this is preserved by substitution (pullback). So in the fibred case, if the column on the right is an equaliser diagram then so is the substituted version on the left. Then E is the equaliser in the whole display category $\mathcal{P} = \mathcal{S}^2$ as well as in the fibre over A.

§**4.1.6** Now we shall look at products in **Set** from the "indexed family" point of view. Like universal and type quantification is corresponds to a right adjoint to substitution. This gives us a notion of "internal product" applicable to categories of retracts and domains.

Given an B-indexed family of sets, $(X_b : b \in B)$, their product has elements the indexed families $(x_b : b \in B)$ where $\forall b.x_b \in X_b$. This is a B-indexed family of choices of elements, which is the same as specifying a B-indexed family of maps from the (constant) singleton to the X_b, *i.e.* a map $1_B \to X$ from the terminal object in the fibre over B. Now the display map of the *terminal* object over B is precisely (as an $\mathcal{S}$-morphism) the *identity* on B (which to some extent excuses the ambiguous notation 1_B) so this is just a *section* or *splitting* of the display map.

Write $\prod_B X$ for the product set, and think of it as an object of the fibre over the terminal object (*i.e.* of the category of single sets). It has elements $1_1 \to \prod_B X$, and these are to correspond to the maps $1_B \to X$ over B. Now 1_B is the pullback of 1_1 against the terminal projection $\alpha = !_B : B \to 1$, *i.e.* its image under the substitution functor $\mathtt{P}\alpha$.

$$
\frac{1_B \cong \; \mathtt{P}\alpha \;\; 1_1 \longrightarrow_B X}{1_1 \longrightarrow_1 \prod\limits_B X}
$$

Thus $\prod_B$ is the right adjoint of $\mathtt{P}\alpha$: $\mathtt{P}\alpha 1_1 \cong 1_B \to X$ over B.

§**4.1.7** Now let us do this *indexedly*. So given $((X_b : b \in B_a) : a \in A)$, an A-indexed family of B_a-indexed families of objects, we need to show how to construct the *ath* product, $(\prod_{b \in B_a} X_b : a \in A)$, and present it as a member of an A-indexed family.

To do this we begin by displaying the B_a's over A, *i.e.* we construct a morphism $\alpha : B \to A$, where $B = \coprod B_a$; then we present the objects $((X_b : b \in B_a) : a \in A)$ as a B-indexed family

$(X_b : b \in B)$. The elements of the member $(\Pi\alpha X)_a = \prod_{b\in B_a} X_b$ of the product are maps to $1 \to (\Pi\alpha X)_a$ which are to correspond to indexed families $(1 \to X_b : b \in B_a)$ and so the maps $1_B \to \Pi\alpha X$ over A correspond to those $1_B \to X$ over B. In other words $\Pi\alpha$ (or α_*) is the right adjoint to the substitution or pullback $\mathsf{P}\alpha$ (or α^*).

$$\frac{1_B \cong \mathsf{P}\alpha\ 1_A \longrightarrow_B X}{1_A \longrightarrow_A \Pi\alpha X \cong \prod_{b\in B_a} X_b : a \in A}$$

Definition An internal *product* in an indexed category is a right adjoint to substitution over a display map.

§**4.1.8** The are no conceptual difficulties in doing this for **Set**: collections of maps may be understood naïvely, and *any* morphism in the base category $\mathcal{S}$ occurs as a display map. This is not so in categories of retracts or domains: we have to make our indexing "continuous", and not every map occurs as a display map.

Likewise the collections of maps (cones) in the definition of product have to be "continuously varying". Whilst clearly **Retr**(Λ) and small categories of domains do not have all products externally, they still "think" they have them in this sense. Where the "continuously varying" is taken to mean computable or definable we have the appropriate restriction on the definitions to make them appropriate to programming or intuitionistic type theory.

§**4.1.9** The case of a product over a constant family in **Set** is very familiar: it is simply an exponential. It essentially follows that a category with indexed products over itself is cartesian closed; moreover the exponentials are preserved by substitution (pullback).

The converse is also true for **Set**. Unfortunately this depends crucially on having *all* finite limits.

Proposition Let $\mathcal{S}$ be left exact. Then the fibration $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ has indexed products iff each fibre is cartesian closed; moreover in this case the exponentials are preserved by pullbacks.

Proof

[⇒] Let $\alpha : Y \to A$ and Z be objects over A, and suppose $\Pi\alpha$ is right adjoint to pullback $\mathsf{P}\alpha$ along α. Then $Z_A^Y = \Pi\alpha(\mathsf{P}\alpha Z)$ is the exponential, for

$$\frac{\dfrac{Y \times_A X \longrightarrow_A Z}{Y \times_A X \cong \mathsf{P}\alpha X \longrightarrow_Y \mathsf{P}\alpha Z \cong Y \times_A Z}}{X \longrightarrow_A \Pi\alpha(\mathsf{P}\alpha Z)}$$

[⇐] Let $\alpha : B \to A$ and suppose that the fibre over A is left exact and cartesian closed. For $\beta : Y \to B$ let $\Pi\alpha Y$ be the equaliser

$$\Pi\alpha Y \hookrightarrow Y_A^B \underset{\mathsf{post}(\beta)}{\overset{\mathsf{K}1_B}{\rightrightarrows}} B_A^B$$

Then for $U \in \mathsf{P}A$, maps $U \to \Pi\alpha Y$ over A correspond to maps $U \to Y_A^B$ whose composites with the above maps are equal, and hence to maps $U \times_A B \to Y$ whose composite with $\beta : Y \to B$ is the right projection, *i.e.* to maps $\mathsf{P}\alpha U \cong U \times_A B \to Y$ over B.

[pres] By preservation we mean that $\mathtt{P}\alpha(Z_A^Y) \cong (\mathtt{P}\alpha Z)_B^{(\mathtt{P}\alpha Y)}$.

$$\begin{array}{r}
U \longrightarrow_B \mathtt{P}\alpha(Z_A^Y) \cong Z_A^Y \times_A B \\
\hline
U \longrightarrow_A Z_A^Y \\
\hline
Y \times_A U \longrightarrow_A Z \\
\hline
Y \times_A U \cong \mathtt{P}\alpha Y \times_B U \longrightarrow_B \mathtt{P}\alpha\, Z \\
\hline
U \longrightarrow_B (\mathtt{P}\alpha Z)_B^{(\mathtt{P}\alpha Y)}
\end{array}$$

□

§**4.1.10** The case for sums is very similar, although we are not allowed to argue in terms of elements any more. Like existential quantification, it corresponds to a left adjoint to substitution.
Let $\alpha : B \to A$ be a display map (with fibres B_a) and $Y \in \mathtt{P}B$. The indexed sum is

$$\Sigma\alpha Y \quad = \quad \left\{\sum\nolimits_{b\in B_a} Y_b : a \in A\right\}$$

Hence a map $\Sigma\alpha Y \to X$ over A consists of maps $Y_b \to X_a$ for $b \in B_a$ (by the definition of a coproduct, for a given $a \in A$ we patch these together to make a map $(\Sigma\alpha Y)_a \to X_a$). The same thing may be presented as a B-indexed family of maps $\{Y_b \to X_{\alpha b} : b \in B\}$, *i.e.* a map $Y \to \mathtt{P}\alpha X$ over B.

Definition An *indexed sum* is a left adjoint to substitution over a display map.
For the cod fibration the following easily-overlooked triviality is appropriate:

Lemma For an $\mathcal{S}$-morphism $\alpha : B \to A$, the pullback functor $\mathtt{P}\alpha : \mathcal{S}/A \to \mathcal{S}/B$ along α has a left adjoint $\Sigma\alpha$ (or $\alpha_!$) given by postcomposition with α. □

Proposition The fibration $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ is cocomplete iff $\mathcal{S}$ has all *finite* (limits and) colimits. □
This is due to Bénabou.

$^+$§**4.1.11** Starting from the indexed approach we now have a direct route to the display map: recall that this was originally given as a disjoint union. Let X be an A-indexed family (object of the fibre over A). It has a terminal projection $X \to 1_A$ in this fibre, and the image of this under the sum functor is of course $\sum_A X \to \sum_A 1_A$ over 1; but $\sum_A 1_A$ is (isomorphic to) A (in the canonical identification of $\mathcal{S}$ with the fibre over 1).

Proposition In the fibration $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$, the fibre over the terminal object is equivalent to $\mathcal{S}$ and the display $X \to A$ corresponds to the map $\sum_A X \to \sum_A 1_A$ in this fibre. □
This is the method by which we shall identify displays of retracts.

§**4.1.12** There is another consequence to having a left adjoint to substitution:

Proposition Let $p : \mathcal{P} \to \mathcal{S}$ be a fibration. Then each substitution functor $\mathtt{P}\alpha$ has a left adjoint iff $p : \mathcal{P}^{op} \to \mathcal{S}^{op}$ is *also* a fibration.
Proof

[⇒] Let $\alpha : B \to A$ in $\mathcal{S}$, Y be over B and Z over C, where $\gamma : A \to C$. Then

$$\frac{\dfrac{\dfrac{Y \longrightarrow_{\alpha;\gamma} Z}{Y \to_B \mathtt{P}\alpha(\mathtt{P}\gamma Z)}}{\Sigma\alpha Y \longrightarrow_A \mathtt{P}\gamma Z}}{\Sigma\alpha Y \longrightarrow_\gamma Z}$$

[⇐] Conversely let $\Sigma\alpha$ be the cosubstitution. □

We shall make use of this in §5.2.

We may form the *vertical opposite*, $\mathcal{P}^{\text{vop}}$, of $\mathcal{P}$, in which we turn round the morphisms *in* the fibres but not *between* them. Then $\mathcal{P}$ has indexed products iff $\mathcal{P}^{op} \to \mathcal{S}^{op}$ is a fibration. Also write $\mathcal{P}^{\text{hop}} \simeq (\mathcal{P}^{\text{vop}})^{op}$; usage of the terms *opfibration* and *cofibration* seems confused.

If both $\mathcal{P} \to \mathcal{S}$ and $\mathcal{P}^{op} \to \mathcal{S}^{op}$ are fibrations then we call them a *bifibration*.

4.2 Fibred Categories

§**4.2.1** The previous section introduced the fibred presentation of **Set** and its internal completeness and cocompleteness. Though our major concern is with categories of types like **Set**, $\mathbf{Retr}(\Lambda)$ or $\mathbf{BiPos}_\mathrm{f}$, we also need to know about how other "large" categories may be presented over the base category of types. This may also be done in terms of fibrations, and the aim of the present section is to relate this to "internal" and "indexed" formulations and to describe common categorical constructions and properties in this new form. In particular in §4.4 we shall prove the various forms of the adjoint functor theorem.

Besides using fibrations to discuss *categories* of types and algebras, in the next chapter we shall apply them to the types (domains) themselves, since domains are posets and hence categories. Conversely we shall find in the next section that **Cat** is indexed over itself with fibrations as displays. This confusion of levels is of course important to the construction of a type of types.

§**4.2.2** As in §4.1.2, write $\mathcal{S}$ for the "base" category, whose objects are the indexing sets and whose morphisms are the relabellings; for the moment this is assumed to be **Set**. Let $\mathcal{C}$ be some (possibly large) category, which we are trying to index over $\mathcal{S}$.

As before, for each set $A \in \mathcal{S}$, we have a category of A-indexed families of objects and morphisms of $\mathcal{C}$; of course naïvely this is just $\mathcal{C}^A$, the A-fold power of $\mathcal{C}$. If $\alpha : B \to A$ is a function we have a functor $\mathcal{C}^\alpha : \mathcal{C}^A \to \mathcal{C}^B$ given by composition. Notice that the assignment $\alpha \mapsto \mathcal{C}^\alpha$ is *contravariant*, so we have a functor $\mathcal{C}^{(-)} : \mathcal{S} \to \mathbf{Cat}^{op}$. More generally this may only preserve identity and composition up to isomorphism, and we have to specify these isomorphisms explicitly (they are called *coherences*).

If $\mathcal{C}$ is a concrete category (so its objects are sets and its morphisms *some* functions between sets) we can copy the construction of §4.1.3 directly to build a fibration.

Lemma Let $\mathcal{C}$ be a concrete category and A a set. Then $\mathcal{C}^A$ is equivalent to the following (concrete) category. The *objects* are disjoint unions $X = \coprod_{a \in A} X_a$ of A-indexed families of objects of $\mathcal{C}$ (*quâ* sets), together with their display maps $X \to A$. The *morphisms* $(X \to A) \to (Y \to A)$ are the functions $X \to Y$ which (i) make the triangle over A commute and (ii), when restricted to the components $X_a \to Y_a$, yield $\mathcal{C}$-morphisms. □

Since the theory of categories is finitary essentially algebraic (§3.2.8), the data $\mathtt{P} : \mathcal{S} \to \mathbf{Cat}^{op}$ yield a *(pre)sheaf of categories*; this may alternatively be viewed as a category in the topos of presheaves on $\mathcal{S}$.

§**4.2.3** For abstract categories we have of course to give an abstract construction. We are aiming for a fibration $p : \mathcal{P} \to \mathcal{S}$, where $\mathcal{P}$ corresponds to $\mathcal{S}^2$ in the case $\mathcal{C} = \mathcal{S}$. For a concrete category, $\mathcal{P}$ has objects over A the $X \to A$ of lemma 4.2.2 and morphisms over $\alpha : B \to A$ the commuting squares which restrict in each component to $\mathcal{C}$-morphisms.

In the abstract case the *objects* of $\mathcal{P}$ over A are the A-indexed families themselves, so $\mathsf{ob}\,\mathcal{P} = \coprod_{A \in \mathcal{S}} (\mathsf{ob}\,\mathcal{C})^A$. The *morphisms* are generated by the vertical ones, *i.e.* those in $\mathcal{C}^A$, together with the horizontal ones over $\alpha : B \to A$, which are provided by *fiat*: one of the form $(X_{\alpha b} : b \in B) \to (X_a : a \in A)$ for each A-indexed family.

Definition

(a) A morphism $f : X \to Y$ in $\mathcal{P}$ is *cartesian* or *horizontal* (w.r.t. $p : \mathcal{P} \to \mathcal{S}$) if, given any $g : Z \to Y$ in $\mathcal{P}$ and a commutative diagram as on the left:

$$\begin{tikzcd} pZ \arrow[d, "\alpha"'] \arrow[dr, "pg"] & \\ pX \arrow[r, "pf"'] & pY \end{tikzcd} \qquad \begin{tikzcd} Z \arrow[d, "h"'] \arrow[dr, "g"] & \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

in $\mathcal{S}$ then there is a unique $h : Z \to X$ in $\mathcal{P}$ with $ph = \alpha$ such that the triangle on the right commutes.

(b) A morphism is *vertical* if its image under p is an identity.

(c) The functor $p : \mathcal{P} \to \mathcal{S}$ is a *fibration* if, given any $\alpha : B \to A$ in $\mathcal{S}$ and $X \in \mathcal{P}$ with $pX = A$, there is a cartesian morphism $f : Y \to X$ in $\mathcal{P}$ with $pf = \alpha$; we call f a *cartesian lifting* of α.

In (c) we only ask for the *existence* of the f, not a *choice* of it (though any two such will be isomorphic). Even so, we shall write $\mathtt{P}\alpha X$ for any such Y.

Proposition Given any category $\mathcal{C}$, the above construction $p : \mathcal{P} \to \mathcal{S}$ is a fibration. If $\mathcal{C}$ is concrete (in particular if $\mathcal{C} = \mathcal{S}$) this is equivalent to the previous construction, in the sense of an equivalence of categories over $\mathcal{S}$. □

Lemma The following diagram is a pullback in $\mathcal{P}$:

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

□

§**4.2.4** We have shown how to pass from the *indexed* to the *fibred* presentation; the inverse is rather more complicated. However before attempting to perform an inverse construction, we observe that there is nothing in this to require the fibre over A to be the A-fold power of $\mathcal{C}$. In fact the foregoing construction depends only on the (pseudo)functor $\mathcal{S} \to \mathbf{Cat}^{op}$.

For a general indexing pseudofunctor $\mathsf{P} : \mathcal{S} \to \mathbf{Cat}^{op}$, we construct a fibration $p : \mathcal{P} \to \mathcal{S}$ as follows. The fibre over A is just $\mathsf{P}A$ (this accounts for the objects and vertical morphisms), and the horizontal morphisms are of the form $\mathsf{P}\alpha X \to X$ over $\alpha : B \to A$ where $X \in \mathsf{P}A$.

There is still something special about the form of the fibration which arises in this way, namely that (contrary to the wording of definition 4.2.3(c)) we have *canonical* choices of cartesian liftings. A fibration of this form is said to be *cloven*. Of course any two cartesian liftings are isomorphic, but there is nevertheless a potential problem of Choice. In the case of $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ this is the distinction between the *existence* and *assignment* of pullbacks.

Whether one regards this as a problem or not depends on context. If, as is Bénabou's view, $\mathcal{S}$-categories and other constructions arise naturally as *arbitrary* fibrations and not as indexed categories, and we aim to work in an arbitrary topos, then clearly this use of Choice is important. In the contexts in which fibrations arise in the present work, on the other hand, which are essentially syntactic, we find that they are naturally derived from indexations anyway. (On the other hand, in the category theory of **Set** we may vary the values of functors at individual objects independently, though of course only up to isomorphism, whereas in domain theory we are obliged to make "continuous" choices.)

Proposition There is an equivalence between pseudofunctors $\mathcal{S} \to \mathbf{Cat}^{op}$ and cloven fibrations over $\mathcal{S}$. The indexation is a functor iff the fibration is *split*, *i.e.* the coherences (between cartesian liftings over identities and composites) are identities. □

The $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ fibration is split iff we have not only a canonical choice of pullbacks, but also the composite (*cf.* §1.2.14) of those for the two squares gives the one for the entire rectangle:

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \arrow[r] & \bullet \end{tikzcd}$$

The question of splitting has algebraic content (*i.e.* apart from questions of Choice) where there are objects with proper automorphisms. We shall therefore look at the case for groups in §4.2.16.

§**4.2.5** So far the only fibred category we have is $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$. Let us therefore construct the fibred category of models of a single-sorted finitary algebraic theory, say rings.

An A-indexed family of rings is of course given by $(R_a : a \in A)$ together with $(\ulcorner 0_a \urcorner, \ulcorner 1_a \urcorner : 1 \to R_a : a \in A)$, $(-_a : R_a \to R_a : a \in A)$ and $(+_a, \times_a : R_a \times R_a \to R_a : a \in A)$. These may be coded as before by $R \to A$, $\ulcorner 0 \urcorner, \ulcorner 1 \urcorner : A \to R$, $- : R \to R$ and $+, \times : R \times_A R \to R$ over A, remembering that a fibred terminal object is the identity and a fibred product is constructed by means of a pullback. The equations are satisfied in the components iff they are for the displays: the reader is invited to formulate the latter condition.

A homomorphism of indexed families of rings is given by the corresponding function on sets (which was just a commutative square), so long as it preserves the operations. Again this happens componentwise iff it happens indexedly.

The reason why the componentwise and indexed formulations coincide is essentially that we have *evaluation functors* $\mathsf{ev}_a : \mathbf{Rng}^A \to \mathbf{Rng}$ for each $a \in A$ which (i) preserve (equations and) products and (ii) are jointly faithful. ev_a is just the substitution functor over the global element $\ulcorner a \urcorner : 1 \to A$.

§**4.2.6** We only so far know about indexed or fibred *categories*: what about functors?

Consider first the indexed case, given two pseudofunctors $\mathsf{P}, \mathsf{Q} : \mathcal{S} \to \mathbf{Cat}^{op}$. Then an indexed functor $\mathbf{t}T : \mathsf{P} \to \mathsf{Q}$ is an assignment of a functor $\mathbf{t}TA : \mathsf{P}A \to \mathsf{Q}A$ to each $A \in \mathcal{S}$. This must be consistent with substitution, so to each $\alpha : B \to A$ we have coherent natural equivalence:

$$\begin{tikzcd}
\mathtt{P}B \arrow[d, "\mathbf{t}TB"'] & \mathtt{P}A \arrow[l, "\mathtt{P}\alpha"'] \arrow[d, "\mathbf{t}TA"] \\
\mathtt{Q}B & \mathtt{Q}A \arrow[l, "\mathtt{Q}\alpha"']
\end{tikzcd}$$

$$B \xrightarrow{\alpha} A$$

[*cf.* the Beck condition. (?)]

Now suppose $p : \mathcal{P} \to \mathcal{S}$ and $q : \mathcal{Q} \to \mathcal{S}$ are the corresponding fibrations. We may collect together the parts $\{\mathbf{t}TA : A \in \mathcal{S}\}$ into a functor $\mathbf{t}T : \mathcal{P} \to \mathcal{Q}$; this preserves fibres so $p = \mathbf{t}T\,;q$ *exactly* (not just up to isomorphism). The cartesian morphisms $\mathtt{P}\alpha X \to X$ in $\mathcal{P}$ over $\alpha : B \to A$ are mapped to $\mathbf{t}TB(\mathtt{P}\alpha X) \cong \mathtt{Q}\alpha(\mathbf{t}TAX) \to \mathbf{t}TAX$ in $\mathcal{Q}$, so $\mathbf{t}T$ preserves cartesian morphisms (though not necessarily choices of them). We call a functor with these properties *cartesian*. Given any cartesian functor between fibrations arising from indexations we may recover the indexed functor.

§**4.2.7** And natural transformations?

Given two indexed functors $\mathbf{t}T, \mathbf{t}U : \mathtt{P} \to \mathtt{Q}$, an indexed natural transformation $\theta : \mathbf{t}T \to \mathbf{t}U$ is an assignment of a natural transformation $\theta A : \mathbf{t}TA \to \mathbf{t}UA$ to each $A \in \mathcal{S}$ consistent with substitution in the sense that the following diagram commutes for each $\alpha : B \to A$.

$$\begin{tikzcd}
\mathbf{t}TB(\mathtt{P}\alpha X) \arrow[r, "\cong"] \arrow[d, "\theta BX"'] & \mathtt{Q}\alpha(\mathbf{t}TAX) \arrow[r, "\lrcorner" description] \arrow[d, "\mathtt{Q}\alpha(\theta AX)"'] & \mathbf{t}TAX \arrow[d, "\theta AX"'] \\
\mathbf{t}UB(\mathtt{P}\alpha X) \arrow[r, "\cong"] & \mathtt{Q}\alpha(\mathbf{t}UAX) \arrow[r] & \mathbf{t}UAX
\end{tikzcd}$$

$$B = B \xrightarrow{\alpha} A$$

This gives rise to a natural transformation $\theta : \mathbf{t}T \to \mathbf{t}U$ of cartesian functors. The above condition is equivalent to naturality w.r.t. horizontal maps, so any such θ will do.

Proposition Let $\mathtt{P}$, $\mathtt{Q}$ be indexations and p, q the corresponding fibrations. Then there is a natural equivalence between indexed and cartesian functors and natural transformations. □

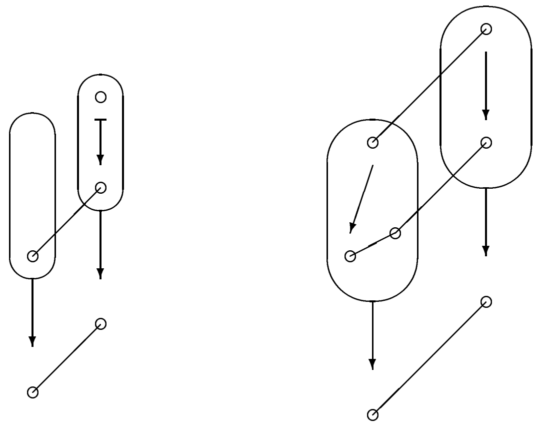
When we speak of functors and natural transformations between fibred categories we shall always mean them to be cartesian.

§**4.2.8** Given a fibred category $p : \mathcal{P} \to \mathcal{S}$, an arbitrary subcategory of $\mathcal{P}$ needn't be fibred over $\mathcal{S}$, and even if it is the inclusion needn't be cartesian.

Examples Posets are sufficient to show this.

(a) A non-fibred subcategory.

(b) A functor between fibrations which doesnt preserve cartesian maps; this is a non-cartesian inclusion of a fibred subcategory.



(a) (b)

Figure 4.2.8: Non-cartesian subcategories

§**4.2.9** We can now speak of indexed adjoints.

Proposition Let $p : \mathcal{P} \to \mathcal{S}$ and $q : \mathcal{Q} \to \mathcal{S}$ be fibrations and $\mathbf{t}T : \mathcal{P} \to \mathcal{Q}$, $\mathbf{t}U : \mathcal{Q} \to \mathcal{P}$ be cartesian functors.

(a) $\mathbf{t}T$ is left adjoint to $\mathbf{t}U$ *quâ* cartesian functors iff they are adjoint *quâ* ordinary functors.

(b) In this case $\mathbf{t}T$ preserves indexed sums and $\mathbf{t}U$ preserves indexed products.

Proof

[a] From §4.2.7 we impose no additional condition on natural transformations to make them cartesian, and likewise the triangle equations for adjoints coincide.

[b] By taking the diagram of right adjoints to §4.2.6, $\mathbf{t}U$ commutes with Π functors, *i.e.* preserves products; likewise $\mathbf{t}T$ preserves sums (Σ). □

§**4.2.10** Having considered "large" categories, let us now turn to "small" or internal ones. We showed in §3.2.8 how to interpret the theory of categories in any category with finite limits. Most of the information is contained in $\langle \mathsf{dom}, \mathsf{cod} \rangle : C_1 \to C_0 \times C_0$; considered as a display this corresponds to the indexed family $(\mathsf{hom}_{\mathcal{C}}(x, y) : x, y \in C_0)$.

The fibred version of this category is given as follows. The *objects* over A are A-indexed families of "objects" of $\mathcal{C}$ ("elements" of C_0), *i.e.* just functions ($\mathcal{S}$-maps) $x : A \to C_0$. The *morphisms* $f : (y : B \to C_0) \to (x : A \to C_0)$ over $\alpha : B \to A$ are functions $f : B \to C_1$ with $f\,; \mathsf{dom} = y$ and $f\,; \mathsf{cod} = \alpha\,; x$.

Notice that we have a *generic object* $1 : C_0 \to C_0$, of which any other is obtained by substitution (precomposition), and also a *generic morphism*, $(1 : C_1 \to C_1) : (\mathsf{dom} : C_1 \to C_0) \to (\mathsf{cod} : C_1 \to C_0)$. We shall return to small categories in §4.3.8 and §5.5.2.

§**4.2.11** We now turn to manipulation of fibrations, and in particular to change of base. However since we shall be making extensive use of them, we had better first find out what a pullback of categories looks like. The peculiar notation is adapted to our applications.

Since **Cat** is a 2-category, we have to modify the notion of pullback. For in the following diagram, the images of the functors p and F may "miss" one another up to *equality* on objects of

$\mathcal{S}$, whilst having *isomorphic* objects in common; in this case the *strict* pullback falls short of the intention. Replacing equality by isomorphism gives *pseudopullback*.

Proposition Let $F : \mathcal{T} \to \mathcal{S}$, $p : \mathcal{P} \to \mathcal{S}$ be functors as in the figure on the left. Then the pseudopullback of F and p in **Cat** has *objects* the triples (A, X, u) with $A \in \mathcal{T}$, $X \in \mathcal{P}$ and $u : FA \cong pX$ in $\mathcal{S}$ and *morphisms* $(\alpha, f) : (B, Y, v) \to (A, X, u)$ where $\alpha : B \to A$ in $\mathcal{T}$, $f : Y \to X$ in $\mathcal{P}$ and the right-hand square commutes.

$$\begin{tikzcd}
\mathcal{T} \times_{\mathcal{S}} \mathcal{P} \arrow[r] \arrow[d] \arrow[dr, phantom, "\overset{u}{\Rightarrow}"] & \mathcal{P} \arrow[d, "p"'] \\
\mathcal{T} \arrow[r, "F"] & \mathcal{S}
\end{tikzcd}
\qquad
\begin{tikzcd}
FB \arrow[r, "v"] \arrow[d, "F\alpha"'] & pY \arrow[d, "pf"] \\
FA \arrow[r, "u"] & pX
\end{tikzcd}$$

Specifically the pullback square commutes up to the natural isomorphism u, where $u_{(A,X,u)} = u$. If we have another pair of functors $r : \mathcal{C} \to \mathcal{T}$, $\mathtt{G} : \mathcal{C} \to \mathcal{P}$ making the square commute up to the natural isomorphism $v : (r\,; F) \to (\mathtt{G}\,; p)$ then the mediating functor $\langle r, \mathtt{G} \rangle : \mathcal{C} \to \mathcal{T} \times_{\mathcal{S}} \mathcal{P}$ takes U to $(rU, \mathtt{G}U, v_U)$ and $\theta : V \to U$ to $(r\theta, \mathtt{G}\theta)$. $\square$

+§**4.2.12** There is nothing in this construction which really requires u and v to be isomorphisms. There is a more general notion, called *lax pullback*, in which they are replaced by arbitrary 2-cells. We write this $\mathcal{T} \times_{\mathcal{S}} \mathcal{P}$. There are in fact applications of lax constructions to problems in the semantics of programming languages (to powerdomains and to the theory of partial functions) but these topics are not treated in this work.

Note The use of the term lax pullback here is incorrect; the standard name is *comma category*, although it does seem regrettable that the name of an important concept is derived from a non-intuitive piece of notation.

The three forms of pullback are related in the case of a fibration.

Proposition In the notation of proposition 4.2.11, let p be a fibration.

(a) The forgetful functor $\mathcal{T} \times_{\mathcal{S}} \mathcal{P} \to \mathcal{T} \times_{\mathcal{S}} \mathcal{P}$ from the strict to the lax pullback has a coreflection (right adjoint postinverse).

(b) This restricts to an equivalence between the strict and pseudo pullbacks.

(c) If $F : \mathcal{T} \to \mathcal{S}$ is also a fibration $q : \mathcal{Q} \to \mathcal{S}$ and these correspond to indexations $\mathtt{P}, \mathtt{Q} : \mathcal{S} \to \mathbf{Cat}^{op}$, the pullback $\mathcal{P} \times_{\mathcal{S}} \mathcal{Q}$ corresponds to the product $\mathtt{P} \times \mathtt{Q}$.

Proof

[a] Define $\mathcal{T} \times_{\mathcal{S}} \mathcal{P} \to \mathcal{T} \times_{\mathcal{S}} \mathcal{P}$ by taking $(A, X, u : FA \to pX)$ to $(A, \mathtt{P}uX)$, where $\mathtt{P}uX \to X$ is a cartesian lifting of u at X. The counit is $(A \to A, \mathtt{P}uX \to X)$.

[b] If u is restricted to be an isomorphism, the counit is also an isomorphism.

[c] Same argument as in §4.1.5 $\square$

§**4.2.13** In §4.1.3 we saw that substitution of an indexing variable corresponds to precomposition of the indexation and pullback of the display. The same is true at the higher level.

Lemma Let $\mathtt{P} : \mathcal{S} \to \mathbf{Cat}^{op}$ be an indexation and $p : \mathcal{P} \to \mathcal{S}$ the corresponding fibration, and let $F : \mathcal{T} \to \mathcal{S}$ be any functor. Put $\mathtt{Q} = (F\,;\mathtt{P}) : \mathcal{T} \to \mathbf{Cat}^{op}$ for the composite indexation and $q : \mathcal{Q} \to \mathcal{T}$ for the corresponding fibration. Then q is the pullback of p along F.

Proof We have only to show that $\mathcal{Q}$ coincides with $\mathcal{T} \times_{\mathcal{S}} \mathcal{P}$ as constructed in proposition 4.2.11, with the simplification that we may consider strict pullbacks. The fibre $\mathtt{Q}A$ over $A \in \mathcal{T}$ is the same as that $\mathtt{P}(FA)$ over $FA \in \mathcal{S}$ and the horizontal maps $\mathtt{Q}\alpha X \to X$ in $\mathcal{Q}$ over $\alpha : B \to A$ in $\mathcal{T}$ correspond to those $\mathtt{P}(F\alpha)X \to X$ in $\mathcal{P}$ over $F\alpha : FB \to FA$. □

This has been phrased in such a way as to show also that the pullback of an arbitrary fibration against a functor is a fibration. We call this *change of base*.

Proposition Let $F : \mathcal{T} \to \mathcal{S}$ be a functor. Then pullback along F gives rise to a 2-functor from fibrations, cartesian functors and natural transformations over $\mathcal{S}$ to those over $\mathcal{T}$. □

We also have a notion of a cartesian functor $\mathtt{G} : \mathcal{P} \to \mathcal{Q}$ over a change of base $F : \mathcal{T} \to \mathcal{S}$. This is either a cartesian functor from $\mathcal{P}$ to $F^*\mathcal{Q}$ over $\mathcal{T}$, or more simply a functor making the square commute (exactly) and taking horizontal morphisms of $\mathcal{P}$ over $\mathcal{T}$ to those of $\mathcal{Q}$ over $\mathcal{S}$.

§**4.2.14** One particularly important example of change of base is where $\mathcal{T}$ is a slice category $\mathcal{S}/A$ and the functor is just the forgetful functor dom or $\mathsf{post}(!_A)$. The pullback is the "fibred category of A-indexed families of objects" (*cf.* §4.1.7).

In this case the pullback $(\mathcal{S}/A) \times_{\mathcal{S}} \mathcal{P}$ has another description, being an example of an *arrow* (or *comma*) *category*. For a full treatment of this construction, which also subsumes slice categories, see Mac Lane [1971], §2.6.

Lemma $(\mathcal{S}/A) \times_{\mathcal{S}} \mathcal{P} \quad \simeq \quad (p \downarrow A)$ over $\mathcal{S}/A$.

Proof $(p \downarrow A)$ has *objects* the pairs $(X, \alpha : pX \to A)$ with $X \in \mathcal{P}$ and *morphisms* $f : (X,\alpha) \to (Y,\beta)$ the $f : X \to Y$ in $\mathcal{P}$ with $\alpha = pf\,;\beta$. This is the same description as the pullback. □

In the case $\mathcal{S} = \mathbf{Set}$, $(X, \alpha : pX \to A)$ is $((X_b : b \in B_a) : a \in A)$, an A-indexed family of B_a-indexed families of $\mathcal{P}$-objects, where $B_a = \alpha^{-1}(a) \subset pX$ and X_b is the fibre of X over $b \in pX$.

More generally, given $\alpha : B \to A$ in $\mathcal{S}$, we have a functor $\mathcal{S}/B \to \mathcal{S}/A$ given by postcomposition with α. This gives rise to a comparison map between the pullbacks $(p \downarrow B)$ and $(p \downarrow A)$ of $\mathcal{P}$ over $!_B = \alpha;!_A$ and $!_A$. The image of this at $(X, \beta : pX \to B)$ is $(X, \beta\,;\alpha)$.

§**4.2.15** The previous result verifies one of the axioms for a class of display maps (§4.3.2). We may easily show the other two.

Proposition

(a) A composite of fibrations is a fibration.

(b) Any functor to the degenerate (singleton) category is a fibration.

Proof To find cartesian liftings we simply lift half way and then the rest. If the codomain of a functor is degenerate, we make no nontrivial requests for cartesian liftings. □

This is the one case in which the fibred presentation is simpler than the indexed form. Taking the composite of fibrations corresponds to forming indexed sums; indeed if $p : \mathcal{C} \to \mathcal{B}$ and $q : \mathcal{B} \to \mathcal{A}$ are fibrations corresponding to indexations $\mathtt{P} : \mathcal{B} \to \mathbf{Cat}^{op}$ and $\mathtt{Q} : \mathcal{A} \to \mathbf{Cat}^{op}$ then the fibre over $A \in \mathcal{A}$ of the composite $p\,;q$ is the part of the display $p : \mathcal{C} \to \mathcal{B}$ over the fibre $\mathtt{Q}A \subset \mathcal{B}$.

$^+$§**4.2.16** An indexation is, roughly speaking, an example of a map $\mathtt{P} : A \to \mathcal{C}^{op}$ in a category $\mathcal{C}$ where $\mathcal{C}^{op}$ itself has the structure of a $\mathcal{C}$-object. Conversely $\mathcal{C}$-objects have a category structure, so $\mathcal{C}$ is a subcategory of $\mathbf{Cat}$. But this makes it a 2-category, so other $\mathcal{C}$-objects also have 2-structure and $\mathcal{C}$ is a 3-category, and so on. It seems to me that the reason for this is the confusion of object and meta-levels implicit in the notion of a type-of-types, and (because objects of categories are to be considered *isomorphic* rather than *equal*) explains the inconsistency of equality with type-of-types (§5.5.9).

Let's look at a few familiar cases. First let $\mathcal{C} = \mathbf{Set} \subset \mathbf{Cat}$ as the discrete categories. Then $\mathtt{P} : A \to \mathbf{Set}^{op}$ is just an A-indexed family of sets, and the fibration is the display of their disjoint union. **Pos** hardly differs in principle from the general case, but that of **IPO** and other categories of domains is of course our main interest: we shall return to it in Chapter V.

Consider, then, the case $\mathcal{C} = \mathbf{Gp}$, where we consider a group to be a category with one object $(*)$ and every morphism invertible. Functors and group homomorphisms coincide.

Lemma Let $p : H \to G$ be a group homomorphism.

(a) p is a fibration iff it is surjective.

(b) Then the fibre over the unique object $*$ of G is the kernel K of p.

(c) The substitution functor $\mathtt{P}g$ over the base morphism $g \in G$ acts on K by conjugation in H.

(d) A splitting for this fibration is equivalent to an inclusion of G as a subgroup of H disjoint from K. □

Proposition

(a) Fibrations of groups correspond to extensions.

(b) A fibration is split iff the extension splits. □

$^{+}$§**4.2.17** This leads us to a categorical proof of the Jordan-Hölder decomposition theorem for groups. Here is another example of the fact that categories occur in Mathematics not just as *collections* of familiar objects, but as the objects *themselves*.

Proposition

(a) Let $1 = U_0 \lhd U_1 \lhd U_2 \lhd \ldots \lhd U_n = G$ and $1 = V_0 \lhd V_1 \lhd V_2 \lhd \ldots \lhd V_m = G$ be two *subnormal series* of subgroups of a group G, *i.e.* in which each term is normal in the next (but not necessarily in G). Then there are *isomorphic refinements*, *i.e.* series U'_{ij} and V'_{ji} (with lexicographic order) such that $U'_{(i-1)m} = U_i = U'_{i0}$, $V'_{(j-1)n} = V_j = V_{j0}$ and $U'_{i(j+1)}/U'_{ij} \cong V'_{j(i+1)}/V'_{ij}$.

(b) The lattice L of normal subgroups of a group is *modular*, *i.e.* for $a, b \in L$, the *intervals* $[a \curlywedge b, a] = \{x : a \curlywedge b \leq x \leq a\}$ and $[b, a \curlyvee b]$ are isomorphic posets.

Proof [a] Each step in the series is a fibration, so the result follows from the fact that a pullback of fibrations is a fibration (lemma 4.2.13). [b] is similar. □

$^{+}$§**4.2.18** Fibrations interact straightforwardly with limits.

Proposition Let $\mathcal{P} = \mathsf{lim}\,\mathcal{P}_i$ and $\mathcal{C} = \mathsf{lim}\,\mathcal{C}_i$ in **Cat** and $p_i : \mathcal{P}_i \to \mathcal{C}_i$ be a family of fibrations compatible with the diagrams in the sense that

$$\begin{tikzcd}
\mathcal{P}_j \arrow[r] \arrow[d, "p_j"] & \mathcal{P}_i \arrow[d, "p_i"'] \\
\mathcal{C}_j \arrow[r] & \mathcal{C}_i
\end{tikzcd}$$

is a cartesian functor for each $i \to j$ in I. Then the mediating functor $p : \mathcal{P} \to \mathcal{C}$ is a fibration.

Proof Let $\alpha : B \to A$ in $\mathcal{C}$ and $X \in \mathtt{P}A \subset \mathcal{P}$. α arises from a compatible family $\alpha_i : B_i \to A_i$ in $\mathcal{C}_i$, similarly X from $X_i \in \mathtt{P}_i A_i$ since $p_i X_i = (pX)_i = A_i$. I claim $(\mathtt{P}_i \alpha_i X_i)$ is a compatible family. This is because the functors in the diagram are fibred and hence preserve the horizontal maps $\mathtt{P}_i \alpha_i X_i \to X_i$. We also have a compatible family of horizontal maps $(\mathtt{P}_i \alpha_i X_i \to X_i)$.

Put $\mathtt{P}\alpha X = (\mathtt{P}_i \alpha_i X_i) \to (X_i) = X$; I claim this is horizontal. For let $f : Y \to X$ with pf factoring through α. The we have $(pf)_i = p_i f_i : p_i Y_i \to A_i$ and so a mediating map $Y_i \to \mathtt{P}_i \alpha_i X_i$. By the uniqueness of this map we have a compatible family, and hence a mediating map $Y \to \mathtt{P}\alpha X$. □

We shall make extensive use of this result in the next chapter.

4.3 Relatively Cartesian Closed Categories

§**4.3.1** In this section we look at cartesian closed categories from a indexed or fibred point of view. The weakest form is where $- \times X$ has a right adjoint for each object X; this corresponds to only allowing "constant families" to be displayed. At the other extreme we have the case of **Set**, where any map occurs as a display and pullback against it has a right adjoint; we call this *local* cartesian closure. We shall find that for domains we need an intermediate concept, which is parametric upon the chosen class of display maps so is called *relative* cartesian closure. We leave the case of domains to the next chapter, but find already here that we can apply this definition to **Cat**.

§**4.3.2** First let us establish the

Definition Let $\mathcal{C}$ be a category and $\mathcal{D}$ a class of $\mathcal{C}$-morphisms. We say that $\mathcal{D}$ is *a class of display maps* for $\mathcal{C}$ if

(i) The pullback of any $\mathcal{D}$-map against any $\mathcal{C}$-map exists and is in $\mathcal{D}$,

(ii) The composite of any two $\mathcal{D}$-maps is in $\mathcal{D}$, and,

(iii) $\mathcal{C}$ has a terminal object and any terminal projection is in $\mathcal{D}$.

Examples

(a) Let $\mathcal{C}$ be any lex category (with all finite limits), *e.g.* **Set**. Then $\mathcal{C}$ is a class of display maps for itself.

(b) Let $\mathcal{C}$ be any category with finite products and $\mathcal{D}$ the class of all (maps isomorphic to) product projections.

(c) Let $\mathcal{C} = $ **IPO** and $\mathcal{D}$ the projections. (Proposition 2.1.8)

(d) Let $\mathcal{C} = $ **Cat** and $\mathcal{D}$ the fibrations. (Proposition 4.2.13 & 15) □

Examples (a) and (b) are the maximal and minimal cases; we shall be interested in the other two, and the relation between them.

+§**4.3.3** Recall that the slice category $\mathcal{C}/A$ has objects the $\mathcal{C}$-morphisms with codomain A and morphisms the $\mathcal{C}$-morphisms making the triangle commute. If $\alpha : B \to A$ is a $\mathcal{C}$-morphism there is a functor $\alpha_! : \mathcal{C}/B \to \mathcal{C}/A$ given by postcomposition with α. The right adjoint to $\alpha_!$, if it exists, is called α^* and is given by pullback along α. (If α^* itself has a right adjoint, written α_*, then α

is said to be *exponentiable*; when we regard α^* as a substitution we write it as $\mathtt{P}\alpha$ and its adjoints as $\Sigma\alpha$ and $\Pi\alpha$.)

Warning "Slice *over*" in a category corresponds to "*down*-set" in a poset.

More generally let $\mathcal{D}$ be a class of $\mathcal{C}$-maps; the *relative slice* $\mathcal{C}/_{\mathcal{D}}A$ has objects the $\mathcal{D}$-maps with codomain A but still *all* $\mathcal{C}$-maps as morphisms (so long as the triangle still commutes in $\mathcal{C}$). Thus $\mathcal{C}/_{\mathcal{D}}A$ is a *full* subcategory of $\mathcal{C}/A$.

Warning This seems *not* to be quite the correct definition, for in the case of categories and fibrations we have the extra condition of preserving cartesian morphisms. I cannot see at the moment what the appropriate morphism-part of this object-definition is, but we do not rely too heavily on the definition in the rest of the work.

[Postscript: it is actually correct.]

From these data we can construct a fibred category $p : \mathcal{P} \to \mathcal{C}$, which is in some sense "$\mathcal{C}$ fibred over itself".

The *objects* over $A \in \mathcal{C}$ are the display maps $X \to A$ with codomain A, and the *morphisms* over $\alpha : B \to A$ from $Y \to B$ to $X \to A$ are the commutative squares of which three sides have already been given. The fibre $\mathtt{P}A$ over A is therefore $\mathcal{C}/_{\mathcal{D}}A$.

Lemma If $(\mathcal{C}, \mathcal{D})$ satisfy axiom (i) above, then this is a fibration; the fibres are relative slices and the horizontal maps are the pullback squares. □

This coincides with the standard construction in the case of **Set**; for the minimal class of display maps the families are all "constant" ones, so this isn't very interesting.

$^+$**§4.3.4** Axiom (ii) serves a dual rôle: it performs mundane categorical bookkeeping, but also provides indexed sums. The purpose of axiom (iii) is that we should be able to speak of the fibration as actually being the indexed form of the original category.

Proposition Let $\mathcal{D}$ be a class of display maps for a category $\mathcal{C}$. Then in the fibration constructed above,

(a) $\mathcal{C}$ is canonically identified with the fibre over its terminal object.

(b) $\mathcal{C}$ has finite products, and their projections (including all isomorphisms) are in $\mathcal{D}$.

(c) The fibres have finite products and these are preserved (up to isomorphism) by arbitrary pullback functors.

Proof

[a] The fibre over 1 has objects the display maps with codomain 1, which are all maps to 1 by axiom (iii). But by the definition of a terminal object this category is (canonically) equivalent to $\mathcal{C}$.

[b] Finite products are given by pullback over the terminal object, and the projections are in $\mathcal{D}$ by axiom (i).

[c] We did this in §4.1.5. Preservation follows from the fact that limits commute with one another. □

$^+$**§4.3.5** We also have indexed sums.

Proposition Let $\mathcal{C}$ be a category with a class of display maps $\mathcal{D}$. Then in the fibration;

(a) Pullback along display maps has a left adjoint.

(b) The Beck condition for sums is satisfied.

(c) The display map in the sense constructed in §4.1.11 coincides with that defining the indexed type.

Proof (a) and (c) follow immediately from the remarks in §4.1.11 and §4.3.3. We have just to prove for the pullback square on the left (in which α and γ are displays) that the diagram on the right commutes up to isomorphism.

$$\begin{tikzcd} D \arrow[r, "\delta"] \arrow[d, "\gamma"'] \arrow[dr, phantom, very near start, "\lrcorner"] & A \arrow[d, "\alpha"] \\ B \arrow[r, "\beta"] & C \end{tikzcd} \qquad \begin{tikzcd} \mathtt{P}D \arrow[d, "\Sigma\gamma"'] & \mathtt{P}A \arrow[l, "\mathtt{P}\delta"'] \arrow[d, "\Sigma\alpha"] \\ \mathtt{P}B & \mathtt{P}C \arrow[l, "\mathtt{P}\beta"] \end{tikzcd}$$

Let $\xi : X \to A$ be a display (so $X \in \mathtt{P}A$). Then the following squares and rectangle are pullbacks (lemma 1.2.14), so the isomorphism holds.

$$\begin{tikzcd} \mathtt{P}\beta(\Sigma\alpha X) \cong \Sigma\gamma(\mathtt{P}\delta X) \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & D \arrow[r, "\gamma"] \arrow[d, "\delta"'] \arrow[dr, phantom, very near start, "\lrcorner"] & B \arrow[d, "\beta"] \\ X \arrow[r, "\xi"] & A \arrow[r, "\alpha"] & C \end{tikzcd}$$

□

$^+$**§4.3.6** It therefore remains to discuss indexed products and exponentials.

Definition A category $\mathcal{C}$ is *cartesian closed relative to* a class of display maps $\mathcal{D}$ if the substitution functors of the fibration constructed in §4.3.3 have right adjoints.

Note This is rather clumsily stated. The products (right adjoints) only exist over display maps, take displays to displays and must satisfy the Beck condition. The latter is not redundant (despite §3.2.17) because the adjoints do not exist for all base morphisms. However Thomas Streicher observed that with the Beck condition the product over a display $\alpha : B \to A$, applied to a display $q : Y \to B$, enjoys its universal property with respect to maps f not just from displays (objects of the relative slice) but from arbitrary objects $p : X \to A$ of the slice:

$$\frac{X \xrightarrow{f}_A \Pi\alpha Y}{\alpha^* X \longrightarrow_B Y}$$

and conversely that if the adjoint $\alpha^* \dashv \Pi\alpha$ has this property then the Beck condition is satisfied.

Examples

(a) **Set**, or any locally cartesian closed category, is cartesian closed relative to all maps.

(b) Any cartesian closed category is cartesian closed relative to product projections. □

We shall show that categories of retracts and domains are relatively cartesian closed in §5.1 and 5.4 respectively. Also the projections provide a class of display maps for **IPO**, they do not do so for other categories of domains, and even **IPO** is not cartesian closed with respect to them (§5.2.8).

$$\begin{tikzcd}
Y' \arrow[rr] \arrow[d] \arrow[ddr] & & Y & \\
\mathrm{P}\alpha' X' \arrow[rr] \arrow[dr] \arrow[dd] & & \mathrm{P}\alpha X \arrow[dr] \arrow[dd] & \\
& X' \arrow[rr] \arrow[dd] & & X \arrow[dd] \\
B' \arrow[rr] \arrow[dr] & & B \arrow[dr] & \\
& A' \arrow[rr] & & A
\end{tikzcd}$$

Figure 4.3.8: Locally small category

§**4.3.7** We may sum up our results in the

Theorem A relatively cartesian closed category gives rise to an indexed category whose fibres are cartesian closed and whose substitution functors preserve this structure. It is complete and cocomplete in the sense that substitution along display maps has adjoints on both sides which satisfy the Beck condition. □

§**4.3.8** We can use these methods to formulate definitions and constructions internally (say in a locally cartesian closed category). This usually takes the form of finding a *generic* construction, of which any other is obtained by substitution (pullback), preferably uniquely.

We shall illustrate this by formulating the idea of a category *having small hom-sets* or being *locally small*. Since cartesian closure is concerned with exponentials, *i.e. sets* of functions, it will not come as a surprise that these are equivalent. We can formulate local smallness as having a generic morphism, *i.e.* one from which any other may be obtained by substitution (pullback).

Thus if $Y \in \mathrm{P}B$ and $X \in \mathrm{P}A$ with $\alpha : B \to A$, by a *generic morphism* from Y to X over α we mean a diagram of the form in the figure, in which the squares (parallelograms) are pullbacks, which is generic in the sense that any other diagram of the same shape (but with $''$ for $'$) is obtained by pulling back this one by a unique $A'' \to A'$.

Proposition

(a) A small category is locally small.

(b) Given an A-indexed family of objects in a locally small category, there is a small category fully embedded with object-of-objects A.

Proof

[a] Let the small category be (C_0, C_1). We have a generic object $\mathsf{id} : C_0 \to C_0$, *i.e.* a ($C_0$-indexed) family of which any other is a pullback. There is also a generic morphism $C_1 \to C_0 \times C_0$ between any pair of generic objects, and substituting any other families for the generic ones we have generic morphisms between any pair of (families of) objects.

[b] Let X be over A. Put $C_0 = A$ and C_1 the hom-set from $\mathtt{P}\pi_0 X$ to $\mathtt{P}\pi_1 X$ over $C_0 \times C_0$. Construct the fibration as in §4.2.10 and the obvious embedding. □

§4.3.9

Proposition $\mathcal{S}$ is locally small iff it is locally cartesian closed.

Proof

[⇐] Suppose Y is exponentiable in its fibre. Put $B' = (\mathtt{P}\alpha X)^Y_B$ and $A' = A \times_B B'$. Then

$$
\begin{array}{c}
Y'' \longrightarrow_{\alpha''} X'' \\
\hline
Y \times_B B'' \cong Y'' \longrightarrow_{B''} \mathtt{P}\alpha'' X'' \cong \mathtt{P}\alpha X \times_B B'' \\
\hline
B'' \times_B Y \longrightarrow_B \mathtt{P}\alpha\ X \\
\hline
B'' \longrightarrow_B (\mathtt{P}\alpha\ X)^Y_B = B'
\end{array}
$$

and it's not difficult to see that the correspondence is obtained by pullback.

[⇒] Conversely if B' is generic then

$$
\begin{array}{c}
B'' \times_B Y \longrightarrow_B X \\
\hline
B'' \times_B Y \longrightarrow_{B''} B'' \times_B X \\
\hline
B'' \longrightarrow_B B' \qquad = X^Y_B
\end{array}
$$

□

§4.3.10 We shall now give part of the the proof that **Cat** is cartesian closed relative to fibrations. [But the morphisms of the relative slices in this case are cartesian not arbitrary functors.]

Specifically we shall construct the fibred category of (cartesian) functors between two fibrations. We do this by first constructing the indexed category and then relying on proposition 4.2.4 to give the fibration.

Given fibrations $p : \mathcal{P} \to \mathcal{S}$ and $q : \mathcal{Q} \to \mathcal{S}$ we need a notion of an A-indexed family of cartesian functors $\mathcal{P} \to \mathcal{Q}$. Generally an A-indexed family of functions is a fibre-preserving function between displays. Recall from §4.2.14 that $(p \downarrow A) \simeq (\mathcal{S}/A) \times_{\mathcal{S}} \mathcal{P}$ is the "fibred category of A-indexed families of $\mathcal{P}$-objects", and that its objects are $(X, \alpha : pX \to A)$. Hence cartesian functors $(p \downarrow A) \to (q \downarrow A)$ over $\mathcal{S}/A$ represent A-indexed families of functors $\mathcal{P} \to \mathcal{Q}$. We take this to be the fibre of $\mathcal{Q}^{\mathcal{P}}$ over A.

The substitution functor $\mathtt{Q}^{\mathtt{P}}\alpha : [p \downarrow A, q \downarrow A] \to [p \downarrow B, q \downarrow B]$ over $\alpha : B \to A$ is defined as follows. Let $\mathbf{t}T : p \downarrow A \to q \downarrow A$ be cartesian over $\mathcal{S}/A$ and take $(X, pX \to A)$ to $(\mathbf{t}T_0(X, pX \to A), pX \to A)$. Then we define $\mathtt{Q}^{\mathtt{P}}\alpha\mathbf{t}T : p \downarrow B \to q \downarrow B$ by letting its value at $(X, pX \to B)$ be $(\mathbf{t}T_0(X, pX \to B \to A), pX \to B)$, and likewise by the same postcomposition in the case or morphisms. Replacing $\mathbf{t}T_0$ by θ gives the prescription for natural transformations.

$^{+}$§**4.3.11** Now we aim to show that this is indeed the exponential. We must show a (natural) equivalence between cartesian functors $\mathcal{P} \times_{\mathcal{S}} \mathcal{R} \to \mathcal{Q}$ and $\mathcal{R} \to \mathcal{Q}^{\mathcal{P}}$ over $\mathcal{S}$. Curiously this seems to have more to do with the *lax* pullback $\mathcal{P} \times_{\mathcal{S}} \mathcal{R}$ than the strict or pseudo one which is directly given, and the reader is advised to keep the adjunction from proposition 4.2.12a in mind.

The first part of the equivalence takes

$$\begin{tikzcd} \mathcal{R} \arrow[rr, "\mathbf{t}T"] \arrow[dr, "r"'] & & \mathcal{Q}^{\mathcal{P}} \arrow[dl, "q^p"] \\ & \mathcal{S} & \end{tikzcd} \quad \text{to} \quad \begin{tikzcd} \mathcal{P} \times_{\mathcal{S}} \mathcal{R} \arrow[r, shift left] \arrow[dr, "p=r"'] & \mathcal{P} \times_{\mathcal{S}} \mathcal{R} \arrow[l, shift left, Rightarrow] \arrow[r, "\mathbf{t}T'"] & \mathcal{Q} \arrow[dl, "q"] \\ & \mathcal{S} & \end{tikzcd}$$

Suppose for $Z \in \mathcal{R}$ over A, $\mathbf{t}TZ : p \downarrow A \to q \downarrow A$ takes $(X, \alpha : pX \to A)$ to $(\mathbf{t}T_0(X, Z, \alpha), \alpha)$. Then we define $\mathbf{t}T'(X, Z, \alpha : pX \to rZ) = \mathbf{t}T_0(X, Z, \alpha)$. It's a bit more complicated on morphisms:

$$\begin{matrix} X \xrightarrow{f} Y \\ \\ Z \xrightarrow{u} W \end{matrix} \qquad \begin{tikzcd} pX \arrow[r, "pf"] \arrow[d, "\text{such that}\,\alpha"'] & pY \arrow[d, "\beta"] \\ rZ \arrow[r, "ru"] & rW \end{tikzcd} \quad \text{goes to} \quad \begin{tikzcd} \mathbf{t}T_0(X, Z, \alpha) \cong \mathbf{t}T_0(X, W, (\alpha \,; ru)) \arrow[d, "\mathbf{t}T_0(f, W)"] \\ \mathbf{t}T_0(Y, W, \beta) \end{tikzcd}$$

The effect of the correspondence on the natural transformation $\theta : \mathbf{t}T \to \mathbf{t}U$ at (X, Z, α) is just $\theta Z(X, \alpha)$.

The foregoing construction does not depend on choices of cartesian liftings; naturally its inverse does. We have to take

$$\begin{tikzcd} \mathcal{P} \times_{\mathcal{S}} \mathcal{R} \arrow[r, shift left, Rightarrow] \arrow[dr, "p=r"'] & \mathcal{P} \times_{\mathcal{S}} \mathcal{R} \arrow[l, shift left] \arrow[r, "\mathbf{t}T'"] & \mathcal{Q} \arrow[dl, "q"] \\ & \mathcal{S} & \end{tikzcd} \quad \text{to} \quad \begin{tikzcd} \mathcal{R} \arrow[rr, "\mathbf{t}T"] \arrow[dr, "r"'] & & \mathcal{Q}^{\mathcal{P}} \arrow[dl, "q^p"] \\ & \mathcal{S} & \end{tikzcd}$$

We must define $\mathbf{t}T$ on objects, vertical morphisms and horizontal morphisms. On the object Z of $\mathcal{R}$ over A we define $\mathbf{t}TZ : p \downarrow A \to p \downarrow A$ by $\mathbf{t}TZ(X, \alpha) = (\mathbf{t}T'(X, \mathtt{R}\alpha Z), \alpha)$ and $\mathbf{t}TZ(f) = \mathbf{t}T(f, h)$ where $f : (X, \alpha) \to (Y, \beta)$ and

$$\begin{tikzcd} \mathtt{R}\alpha Z \arrow[rr, "h"] \arrow[dr] & & \mathtt{R}\beta Z \arrow[dl] \\ & Z & \end{tikzcd} \quad \text{is the cartesian lifting of} \quad \begin{tikzcd} pX \arrow[rr, "pf"] \arrow[dr, "\alpha"'] & & pY \arrow[dl, "\beta"] \\ & A & \end{tikzcd}$$

On the vertical morphism $u : Z \to W$ over $A = rZ = rW$ we have to define a natural transformation $\mathbf{t}Tu : \mathbf{t}TZ \to \mathbf{t}TW$. The value of this at (X, α) is just $\mathbf{t}T(X, \mathtt{R}\alpha u)$. On the horizontal morphism $\mathtt{R}\alpha Z \to Z$ over $\alpha : B \to A$ we have essentially only to observe that the square

$$\begin{tikzcd} p \downarrow B \arrow[r, "\mathbf{t}T(\mathtt{R}\alpha Z)"] \arrow[d] & q \downarrow B \arrow[d] \\ p \downarrow A \arrow[r, "\mathbf{t}TZ"] & q \downarrow A \end{tikzcd}$$

commutes up to isomorphism. On natural transformations $\theta : \mathbf{t}T \to \mathbf{t}U$ we have to define the effect at the object Z of $\mathcal{R}$ over A; this is a natural transformation $\theta_Z : \mathbf{t}TZ \to \mathbf{t}UZ$ defined at (X, α) by $\theta_{(X, \mathtt{R}\alpha Z)}$.

Proposition $\mathcal{Q}^{\mathcal{P}}$ as defined above is the exponential in the fibre of $\mathbf{Cat}/_{\text{fib}}$ over $\mathcal{S}$.

Proof We observe equivalence and naturality by the form of the above construction. $\square$

§**4.3.12** As a by-product of this construction we can assign to any fibration $p : \mathcal{P} \to \mathcal{S}$ an *associated split fibration* $p' : \mathcal{P}' \to \mathcal{S}$, which is the fibred category of (cartesian) functors from $1 : \mathcal{S} \to \mathcal{S}$ to $p : \mathcal{P} \to \mathcal{S}$. There is a cartesian functor $\mathcal{P}' \to \mathcal{P}$ which is full and faithful; with Choice it is essentially surjective on objects. A splitting for p is precisely a choice of preinverse.

4.4 The Adjoint Functor Theorem and Toposes

§**4.4.1** This section is a slight diversion from the theme of the use of indexed category theory for the semantics of polymorphism. Its purpose is to illustrate the way in which the formulation of the adjoint functor theorems for indexed categories leads naturally to the notion of an elementary topos. No detailed proofs will be given. This is in turn related *via* the notion of bounded morphism to Grothendieck toposes, which we introduced in §3.3 to classify geometric theories.

§**4.4.2** The adjoint functor theorem for posets (complete lattices) is a triviality which is subsumed in the confusion of terminology between "supremum" and "least upper bound", for we construct the former precisely as the latter. More generally, a monotone function $f : X \to Y$ between complete lattices has a left adjoint iff it preserves infs. The corresponding result for categories, if it is to exist at all, is necessarily more complicated, since we cannot ask for *all* limits because of Freyd's paradox (proposition 1.5.8).

Let $p : \mathcal{P} \to \mathcal{S}$ and $q : \mathcal{Q} \to \mathcal{S}$ be fibred categories and $\mathbf{t}T : \mathcal{P} \to \mathcal{Q}$ a cartesian functor. Let us assume that $\mathcal{P}$ is locally small (§4.3.8) and complete (§4.1.7), and that F preserves limits in the sense of commuting with Π functors (proposition 4.2.9). We aim to formulate and prove the *General Adjoint Functor Theorem* (see Mac Lane [1971] theorem 5.6.2) for these data.

For each object $Y \in \mathtt{Q}A$ we want to construct an initial object for the arrow category $(Y \downarrow F)$. We make this into a fibred category over $\mathcal{S}/A$ in the manner of lemma 4.2.14. The fibre over $\alpha : B \to A$ has objects the diagrams

$$\begin{tikzcd}
\mathtt{Q}\alpha Y \arrow[r] \arrow[d] & Y \\
FX &
\end{tikzcd}$$

$$B \xrightarrow{\alpha} A$$

for $X \in \mathtt{P}B$.

Lemma $(Y \downarrow F)$ is complete. (*cf.* proposition 4.4.10a) $\square$

An initial object for $(Y \downarrow F)$ assigns to each $\alpha : B \to A$ an initial object $X = \mathtt{G}_\alpha Y$ such that $\mathtt{G}_\alpha Y \to \mathtt{G}_1 Y$ is cartesian. $\mathtt{G} = \mathtt{G}_1$ is then the required left adjoint.

$$
\begin{tikzcd}
\mathtt{Q}\alpha Y \arrow[r] \arrow[d] \arrow[dr, phantom, "1"] & Y & \mathtt{Q}\beta Y \arrow[r] \arrow[d] \arrow[dr, phantom, "2"] & Y & \mathtt{P}\gamma X \arrow[r] \arrow[d] \arrow[dr, phantom, "3"] & X \\
FX & & FX' & & X' & \\
B \arrow[r, "\alpha"] & A & C \arrow[r, "\beta"] & A & C \arrow[r, "\gamma"] & B
\end{tikzcd}
$$

$$
\begin{tikzcd}
& \mathtt{Q}\beta Y \arrow[r] \arrow[d] & \mathtt{Q}\alpha Y \arrow[r] \arrow[d] & Y \\
\mathtt{Q}\gamma(FX) \cong & F(\mathtt{P}\gamma X) \arrow[r] \arrow[d] \arrow[dr, phantom, "4"] & FX & \\
& FX' & & \\
& C \arrow[r, "\gamma"] & B \arrow[r, "\alpha"] & A
\end{tikzcd}
$$

Figure 4.4.4: The solution set condition

§**4.4.3** The point of the General Adjoint Functor theorem is to formulate some condition which manufactures this *initial* object (a colimit) from the available *limits*, analogously to finding the *least* upper bound as the *intersection* of *all* upper bounds.

Given a fibred category $\mathcal{C} \to \mathcal{S}$, we say $\mathcal{C}$ is *initially small* if there's some $X \in \mathcal{C}$ such that for any $Y \in \mathcal{C}$ there's a "family" of maps from X to Y, *i.e.* a diagram of the form

$$
\begin{tikzcd}
\bullet \arrow[r] \arrow[d] & X & \\
\bullet \arrow[rr] & & Y
\end{tikzcd}
$$

in which the horizontal maps are cartesian.

Lemma If $\mathcal{C}$ is complete, locally small and initially small then it has an initial object. $\square$

§**4.4.4** This enables us to formulate the General Adjoint Functor Theorem.

Theorem Let $F : \mathcal{P} \to \mathcal{Q}$ be a cartesian functor between locally small fibred categories such that $\mathcal{C}$ has and F preserves indexed limits. Then F has a left adjoint $\mathtt{G}$ iff for every $Y \in \mathtt{Q}A$ there's a diagram (1) such that for any similar diagram (2) there's some diagram (3) such that (4) commutes. $\square$

This is the indexed form of the *solution set condition*.

§**4.4.5** Though the general adjoint functor theorem is "iff", it is rarely of practical value. A more useful result is obtained by making the category look concrete; though the additional conditions quite commonly hold, they do not always: the category **Loc** of locales and continuous maps is not, for instance, well-powered (see Johnstone [1983] §2.2.11). The ideas are also, unfortunately, inapplicable to domain theory.

In an ordinary category $\mathcal{C}$, a *subobject* of an object X is an isomorphism class of monos into X in $\mathcal{C}/X$. We say that $\mathcal{C}$ is *well-powered* if every object has a *set* of subobjects.

In a fibred category $p : \mathcal{P} \to \mathcal{S}$ we have to test being mono against a *family* of maps. So $i : X \to Y$ in $\mathsf{P}A$ is *vertically mono* if $\mathsf{P}\alpha i$ is mono in the (ordinary) category $\mathsf{P}B$ for any $\alpha : B \to A$. Equivalently i is mono in the display $\mathcal{P}$ with respect to any pair of maps over the *same* base map. An arbitrary map $i : X \to Y$ over $\alpha : B \to A$ in $\mathcal{P}$ is vertically mono if its vertical part $X \to \mathsf{P}\alpha Y$ is.

To formulate well-poweredness we need a notation for the set ($\mathcal{S}$-object) of vertical monos into X over A. This is a display map $\mathsf{Mono}_A(X) \to A$ in $\mathcal{S}$. Over this lies the *generic mono* $U \to X$, which is to be vertically mono and such that any other vertical mono $V \to X$ over $\alpha : B \to A$ is obtained by substitution over some $B \to \mathsf{Mono}_A(X)$. We say $\mathcal{C}$ is *weakly* or *exactly well-powered* according as we have saturation (mere existence of such a substitution) or universality (unique existence); in the latter case we write $\mathsf{Sub}_A(X)$ for $\mathsf{Mono}_A(X)$.

§**4.4.6** When is $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ (exactly) well-powered? If so, then for every $X \in \mathcal{S}$ we have an object PX and a subobject $(\in_X) \subset X \times PX$ such that for any subobject $U \subset X \times A$ there's a (unique) $\ulcorner U \urcorner : A \to PX$ making

$$\begin{tikzcd} U \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & (\in_X) \arrow[d] \\ X \times A \arrow[r, "1_X \times \ulcorner U \urcorner"] & X \times PX \end{tikzcd}$$

a pullback.

Definition An *elementary topos* is an exactly well-powered category $\mathcal{S}$ with finite limits. □

Proposition Let $\mathcal{S}$ be a topos.

(a) If $A \in \mathcal{S}$ then $\mathcal{S}/A$ is also a topos.

(b) $\mathcal{S}$ is cartesian closed.

(c) $\mathcal{S}$ is locally cartesian closed.

(d) $\mathcal{S}$ has finite colimits which are stable under pullback; in particular a strict initial object and disjoint coproducts.

Proof (sketches)

[a] This involves coding the inclusion relation between subobjects.

[b] We code the function $f : X \to Y$ by its graph $(1_X, f) : X \subset X \times Y$ and hence the corresponding subobject of $X \times Y$.

[c] follows from (a) and (b)

[d] This was the first result in the huge range of technology for elementary toposes. It can be done within a suitable powerset, since the singleton map may be shown to be mono and the powerset is an internally complete lattice. □

§**4.4.7** A more commonly seen definition of an elementary topos involves the *subobject classifier*, Ω, which is the powerset of 1 in the terminology of the previous section. This has a global element $true : 1 \to \Omega$ with the property that any mono $U \subset X$ is obtained as the pullback along a unique $\ulcorner U \urcorner : X \to \Omega$.

Example Set is a topos in which $\Omega = \{true, false\}$; the function $X \to \Omega$ is the *characteristic function* of the subset U, which takes the value *true* on U and *false* outside. □

The powerset PX of an arbitrary object X is then Ω^X.

Proposition A lex category $\mathcal{S}$ is a topos iff it is cartesian closed and has a subobject classifier. □

Warning We need equalisers as well as products!

Corollary If $\mathcal{S}$ is a topos, the fibration $\mathsf{cod} : \mathcal{S}^2 \to \mathcal{S}$ is complete, cocomplete, locally small, well-powered and co-well-powered.

Proof For the last part, the quotients of an object X correspond to those subobjects of $X \times X$ which are equivalence relations (this is another "exactness" property of toposes). □

§**4.4.8** Some examples of elementary toposes

Proposition Any Grothendieck topos, and hence any category of sheaves on a space or site, is an elementary topos.

Proof See Johnstone [1977], proposition 1.12. □

Example Elementary but not Grothendieck toposes.

(a) The category $\mathbf{Set}_f$ of finite sets.

(b) The Effective Topos [Hyland 1982]. □

§**4.4.9** To return to the adjoint functor theorem, we need the indexed form of a *cogenerator* (see §1.2.10 and §2.3.2). This is an object $G \in \mathtt{P}A$ such that given any pair of maps $f, g : X \rightrightarrows Y$ over the same $C \to B$ such that for any $h : Y \to G$ we have $f\,;h = g\,;h$ then already $f = g$.

Lemma Let $\mathcal{C} \to \mathcal{S}$ be a complete, locally small, well-powered category with a cogenerator; then $\mathcal{C}$ has an initial object. □

Theorem (Special Adjoint Functor Theorem) Let $F : \mathcal{P} \to \mathcal{Q}$ be a cartesian functor between locally small fibred categories such that $\mathcal{P}$ has and F preserves indexed limits. Suppose $\mathcal{P}$ is well-powered and has a cogenerator. Then F has a left adjoint. □

§**4.4.10** Returning to toposes, suppose we have a functor $\mathbf{t}T : \mathcal{E} \to \mathcal{S}$; what properties on $\mathbf{t}T$ do we need in order to get a well-behaved relationship between fibred categories over $\mathcal{E}$ and $\mathcal{S}$?

From proposition 4.2.13, we have a 2-functor $\mathbf{t}T^*$ taking $\mathcal{S}$-categories to $\mathcal{E}$-categories by pullback.

Proposition Let $\mathbf{t}T : \mathcal{E} \to \mathcal{S}$ be a functor between lex categories.

(a) If $\mathbf{t}T$ preserves pullbacks and $\mathcal{P} \to \mathcal{S}$ has indexed (co)-products then so does $\mathbf{t}T^*\mathcal{P}$.

(b) If $\mathbf{t}T^*(\mathcal{S}^2)$ has indexed coproducts then $\mathbf{t}T$ preserves pullbacks.

(c) If $\mathbf{t}T$ has a right adjoint then $\mathbf{t}T^*$ preserves local smallness.

(d) If $\mathcal{S}$ is locally cartesian closed then $\mathbf{t}T^*(\mathcal{S}^2)$ is locally small iff $\mathbf{t}T$ has a right adjoint.

(e) If $\mathbf{t}T$ has a right adjoint the $\mathbf{t}T^*$ preserves exact well-poweredness.

(f) If $\mathbf{t}T$ preserves 1 and has a right adjoint $\mathtt{R}$ then $\mathtt{R}$ extends to a cartesian functor $(\mathcal{S} \downarrow \mathbf{t}T) \to \mathcal{E}^2$ over $\mathcal{S}^2$. □

Definition A *geometric morphism* $f : \mathcal{S} \to \mathcal{E}$ between toposes is a pair of functors $f^* : \mathcal{E} \to \mathcal{S}$ and $f_* : \mathcal{S} \to \mathcal{E}$ such that f_* is right adjoint to f^* and the latter preserves finite limits.

Examples

(a) Let $f : X \to Y$ be a continuous map between (sober) spaces. Then f gives rise to a geometric morphism $f : \mathbf{Shv}(X) \to \mathbf{Shv}(Y)$. Moreover there is an equivalence between the poset of lemma 2.1.5e and the category of geometric morphisms.

(b) Let M be a model of a geometric theory $\mathcal{T}$ in a topos $\mathcal{E}$. Then there is a geometric morphism $\ulcorner M \urcorner : \mathcal{E} \to \mathbf{Set}[T]$ with $\ulcorner M \urcorner^* T \cong M$. Indeed there is an equivalence as in §3.3.9.

(c) Let $\mathcal{S}$ be a topos and $\alpha : B \to A$ in $\mathcal{S}$. Then pullback $\alpha^* : \mathcal{S}/A \to \mathcal{S}/B$ preserves pullbacks but not 1 and has a right adjoint.

In fact any functor which preserves pullbacks and has a right adjoint factors as the inverse image f^* of a geometric morphism and a functor of this form.

§**4.4.11** What is the relationship between Grothendieck and elementary toposes? Given any Grothendieck topos $\mathcal{E}$, we have a geometric morphism $\mathcal{E} \to \mathbf{Set}$, since $\mathbf{Set}$ is the (pseudo) terminal object in the category of Grothendieck toposes; for sheaves on a space X this corresponds to the terminal projection $X \to 1$.

A geometric morphism $f : \mathcal{E} \to \mathcal{S}$ is *bounded* if $\mathcal{E}$ has an object of generators over $\mathcal{S}$.

Theorem (Giraud, Mitchell & Diaconescu) An elementary topos $\mathcal{E}$ is Grothendieck iff it has a bounded geometric morphism to $\mathbf{Set}$. □

$^+$§**4.4.12**

Question Is a weakly well-powered, locally cartesian closed category necessarily a topos?

We have an Ω which is a "saturated" but not "universal" mono. There is an $\mathcal{S}$-indexed Heyting prealgebra of monos into (rather than subobjects of) each object, which has $\mathcal{S}$-indexed meets. Moreover pullback preserves this structure. It may be possible to apply the adjoint functor theorem internally to construct joins and show that they are preserved. This would give a tripos, but in fact we only really need stable effective image factorisations to construct the "real" (*i.e.* universal) Ω.

Answer Martin Hyland has pointed out that the $\neg\neg$-separated objects of the Effective Topos provide a counterexample.

Chapter 5

Polymorphism in Domains

5.1 Indexed Category of Retracts

$^{+}$§**5.1.1** In this final chapter we shall bring together our studies of Domain Theory and Polymorphism, making use of the Indexed Category Theory of the previous chapter. The first half of the chapter is devoted to Indexed Domain Theory, which we find to be a very pleasing structure. The second half discusses the notion of a "type-of-types" in the context of indexed category theory and attempts to construct such a device for domains, although we find that what can be achieved is rather disappointing.

We motivate our definitions by recourse to the category of retracts of a combinatory algebra; recall that one of the objectives of Chapter II was to show that we may as well assume any category of domains to be of this form. We begin with a very natural definition of an indexed family of retracts and develop the indexed version of the theory of §1.3.3-6. Then we show that global sums and products may be constructed, and hence follow the "display" idea of §4.1.3 to define display maps; we show that indexed sums and products exist relative to these.

The purpose of studying the retract case is that it very naturally motivates a definition for domains. This definition seems to have rather a lot of clauses, so we spend the second section attempting to drop them. Having thereby shown that the variants yield a much inferior theory, we show in the third section that the correspondence between indexed and fibred holds nicely for various categories of domains and in the fourth that we have indexed products. This completes the indexed version of standard domain theory.

In §5.5 we examine the notion of a type of types in an indexed category, showing in particular that we cannot have this together with *both* equality and function-spaces. Dropping the requirement for equalisers (which we have needed to do anyway in domain theory) we introduce a provisional definition of a "typos" and show that the familiar closure model (§1.4.8) provides a nontrivial example.

In §5.6 we construct a "type-of-types" according to this definition for some categories of domains (specifically $\mathbf{ContLat}_\omega$ and $\mathbf{bcCont}_\omega$, but it would probably be possible to do it for others). We can tie this in with §5.1 by showing that we may choose a λ-model in such a way that any fibred type (*quâ* domain) occurs as a display (*quâ* retract). This completes the motivation from the retract to the domain case. Pushing this a little further allows us to code up $\mathsf{G} \to \mathsf{V}$ as a single combinator $\mathsf{V} \in \Lambda$.

We are then in a position to interpret polymorphism in domain theory. Unfortunately we find that $\forall X.X \to X$ does not have its expected value of 2, and that such failure is inevitable in classical poset domain theory. We conclude with a few speculations as to how some of these problems may be resolved.

$^{+}$§**5.1.2** Since a type is an idempotent element, an indexed or parametric type is a function or program which always returns idempotent values. Thus an A-indexed type is a function $X : A \to$

$\mathsf{Idem}(\Lambda)$, where $\mathsf{Idem}(\Lambda)$ is the subset of Λ consisting of the idempotent elements. We force X to have domain A by imposing the equation $X = \mathsf{P}AX$.

Now Ershov showed [Hosono and Sato 1977] that $\mathsf{Idem}(P\omega)$ is a complete lattice but not continuous, and so it is not a type (object of $\mathbf{Retr}(P\omega) \simeq \mathbf{ContLat}_\omega$). But this does not matter, because we do not need a type-of-types to formulate this notion. We only need $X : A \to \Lambda$ to satisfy

$$\forall a \in \|A\| \quad Xa = \mathsf{P}(Xa)(Xa)$$

The left-hand side of this equation is $\lambda u.Xa(Xau)$. It is convenient to introduce a variant on P to deal with this form (and the corresponding one for indexed functions):

$$\mathsf{Q} = \lambda fgau.ga(fau)$$

so that our equation becomes $X = \mathsf{Q}XX$ as functions of a variable $a : A$. These two equations are formally equivalent if Λ is a *model* (has enough points), but in any case we really want the latter because we had *intentional* rather than *extensional* equality in mind anyway.

Hence an A-indexed type is an element $X \in \Lambda$ satisfying

$$X = \mathsf{P}AX = \mathsf{Q}XX$$

By a similar argument, an A-indexed function from Y to X is an $f \in \Lambda$ such that $f = \mathsf{P}Af = \mathsf{Q}Yf = \mathsf{Q}fX$. We then have a cartesian closed category with fixpoints, which we shall call $\mathtt{P}A$.

$^+$§**5.1.3** $\mathtt{P}A$ has *terminal object* $\mathsf{U} = \mathsf{K}(\mathsf{K}\bot)$, and this also represents the *terminal projection* $X \to \mathsf{U}$. As before we may consider (A-indexed) elements: $x \in_A X$ is a function $x : A \to \Lambda$ such that $xa \in Xa$ for each $a \in A$; the type of all such indexed elements of X is the (global) product, $\prod_A X = \lambda ua.Xa(u(Aa))$ (see below).

The *product* $X \times_A Y$ of X and Y over A is given by $\lambda az.\langle Xaz_0, Yaz_1\rangle$; the *projection* maps are $\pi_0 = \lambda az.Xaz_0$ and $\pi_1 = \lambda az.Yaz_1$, and if $f : Z \to X$, $g : Z \to Y$ are two maps in $\mathsf{P}A$ then the *pair* is $\langle f, g\rangle = \lambda az.\langle faz, gaz\rangle : Z \to X \times_A Y$.

We have fibred *exponential* types.

$$\begin{aligned} Y_A^X &= \lambda af.\mathsf{P}(Xa)(\mathsf{P}(fa)(Ya)) \\ &= \lambda af.\mathsf{Q}(\mathsf{Q}Yf)Za \\ &= \lambda af.\mathsf{Q}Y(\mathsf{Q}fZ)a \\ &= \lambda afx.Ya(fa(Xax)) \end{aligned}$$

This is obtained by an interchange of variables from the reduction of f to a solution of $f = \mathsf{P}Af = \mathsf{Q}Xf = \mathsf{Q}fY$; the latter is a "global section" of the former. The *evaluation map* $\mathsf{ev} : Y_A^X \times_A X \to Y$ is given by $\lambda au.Ya(u_0(Xau_1))$ and the *transpose* identifies $f : W \times_A X \to Y$ with $g : W \to Y_A^X$ by $g = \lambda awx.fa\langle w, x\rangle$ and $f = \lambda au.gau_0u_1$.

$^+$§**5.1.4** Now let $\alpha : B \to A$ be any map in the base category $\mathbf{Retr}(\Lambda)$; what is the corresponding *substitution functor* $\mathtt{P}\alpha : \mathtt{P}B \to \mathtt{P}A$, and does it have adjoints? The first question has an easy answer, which gives a pleasing consonance of notation: $\mathtt{P}\alpha = \mathsf{P}\alpha$. In the same way as the naïve indexing of **Set** over itself performed substitution by composition, so does this.

Thus $\mathtt{P}\alpha X$ is simply $\mathsf{P}\alpha X$ and $\mathtt{P}\alpha f = \mathsf{P}\alpha f$; moreover $\mathsf{P}\alpha\mathsf{U} = \mathsf{U}$, $\mathsf{P}\alpha(X \times_A Y) = (\mathsf{P}\alpha X) \times_B (\mathsf{P}\alpha Y)$ and $\mathsf{P}\alpha(Y_A^X) = (\mathsf{P}\alpha Y)_B^{(\mathsf{P}\alpha X)}$ *exactly*. The product projections, pairings, evaluation maps and transposes are also preserved exactly. Because of this notational coincidence (which I hope justifies the switch of variables in the combinator P to the even the most uncompromising users of left-handed notation), $\mathtt{P}A$ and $\mathtt{P}\alpha$ will in future be written $\mathsf{P}A$ and $\mathsf{P}\alpha$ respectively.

+§**5.1.5** We have to show that (some of) the substitution functors have adjoints. In fact those that do are called *display maps*, and these arise from global sums. Therefore we first construct global sums, then identify the display maps and finally construct the indexed sums and products.

Let $Y \in \mathsf{P}B$ be a B-indexed family of retracts; write $\prod_B Y$ and $\sum_B Y$ for their *global* product and sum (coproduct), which are objects of $\mathcal{C}$, the base category. [Later we shall construct $\Pi\alpha Y$ and $\Sigma\alpha Y$, the adjoints to substitution over $\alpha : B \to A$; these give objects of the fibre over A.]

Let $B \in \mathcal{C}$ and $Y \in \mathsf{P}B$. The basic idea of $\prod_B Y$ is the set

$$\{u : B \to \Lambda : \forall b. ub \in Yb\}$$

As with indexed families of objects we can code this set up using combinators. To say that u is a function from B we just need $u = \mathsf{P}Bu$ as before. Again as before the condition $ub \in Yb$ can be written as $ub = Yb(ub)$. In this we spot a rare natural occurrence of the S combinator, and again replace $\forall$ by λ. Hence $\prod_B Y$ is the set of solutions of

$$u = \mathsf{P}Bu = \mathsf{S}Yu$$

These are commuting idempotents (given that $B = \mathsf{P}BB$ and $Y = \mathsf{P}BY = \mathsf{Q}YY$), so their composite gives the required type.

Likewise $\sum_B Y$ is based on

$$\{\langle b, y\rangle : b \in B \wedge yb \in Yb\}$$

which is the set of solutions of

$$u = \langle Bu_0, Yu_0u_1\rangle$$

+§**5.1.6** We may now define a *display map* in $\mathbf{Retr}(\Lambda)$ to be (the composite of an invertible followed by) a map of the form $\pi_0 : \sum_A Y \to A$ where $Y \in \mathsf{P}A$. π_0 is in fact $\lambda u.Bu_0$.

We shall take as read a number of trivial properties of pullbacks, including the fact that this definition allows invertibles to be "passed through" display maps.

Lemma (a) The pullback of a display map exists and is a display map.

Proof Let $\alpha : B \to A$ in $\mathcal{C}$ and $X \in \mathsf{P}A$. Put $Y = \mathsf{P}\alpha X$; then the following is a pullback square:

$$\begin{tikzcd}
\sum_B Y = \lambda v.\langle Bv_0, Yv_0v_1\rangle \arrow[r, "\lambda v.\langle \alpha v_0, Yv_0v_1\rangle"] \arrow[d, "\pi_0"'] \arrow[dr, phantom, "\lrcorner", very near start] & \lambda u.\langle Au_0, Xu_0u_1\rangle = \sum_A X \arrow[d, "\pi_0"] \\
B \arrow[r, "\alpha"] & A
\end{tikzcd}$$

Given any other $\beta : C \to B$ and $\gamma : C \to \sum_A X$ making the square commute, the pair is $\lambda c.\langle \beta c, (\gamma c)_1\rangle$. $\square$

Lemma (b) Any terminal projection is a display map.

Proof Given $B \to \mathsf{T}$, put $X = \mathsf{K}B$ and $A = \mathsf{T}$. Then $X \in \mathsf{P}A$ and $\sum_A X = \lambda p.\langle \perp, Bp_1\rangle \cong B$. $\square$

+§**5.1.7** We have already seen that an indexed category has indexed sums so long as it has a class of display maps, *i.e.* closed under composition.

Proposition

(a) A composite of display maps is a display map.

(b) Substitution over a display map has a left adjoint.

Proof

[a] Given $X \in \mathsf{P}A$ and $Y \in \mathsf{P}B$ where $B = \sum_A X$ we want to construct $Z \in \mathsf{P}A$ with $\sum_B Y \cong \sum_A Z$ over A. Put $Z = \lambda au.\langle Xau_0, Y\langle a, u_0\rangle u_1\rangle$; then $i : \sum_A Z \to \sum_B Y$ and $j : \sum_B Y \to \sum_A Z$ are mutually inverse where

$$\begin{aligned} i &= \lambda z.\langle\langle Az_0, Xz_0z_{10}\rangle, Y\langle z_0, z_{10}\rangle z_{11}\rangle \\ j &= \lambda y.\langle Ay_{00}, \langle Xy_{00}y_{01}, Yy_0y_1\rangle\rangle. \end{aligned}$$

[b] $\Sigma\alpha Y = Z$. Indeed $\Sigma\alpha = \lambda Yau.\langle Xau_0, Y\langle Aa, Xau_0\rangle u_1\rangle$. The adjunctive correspondence between $f : \Sigma\alpha Y \to V$ and $g : Y \to \mathsf{P}\alpha V$ is given by $fau = gby$ where $V \in \mathsf{P}A$, $\alpha b = a \in A$, $b \in Xa \subset B$, $y \in Yb$ and $u = \langle b, y\rangle$. The unit $\eta : Y \to \mathsf{P}\alpha(\Sigma\alpha Y)$ and counit $\epsilon : \Sigma\alpha(\mathsf{P}\alpha V) \to U$ are given by reducing $\lambda by.\langle b, y\rangle$ and $\lambda au.u_1$ to appropriate types. $\square$

We have presented $\Sigma\alpha$ as a combinator. In some sense Σ is also a combinator (as we have pretended by our notation), but its expression involves X, the indexed family corresponding to the display α, rather than α itself. This is because $\Sigma\alpha$ only exists when α is a display map (as is easily deduced from the above proof), and in this highly constructive world we need a concrete proof of every necessary fact, which proof is supplied by the X corresponding to α.

Question Is there a combinatorial (in the sense of the definition of X as an A-indexed type) description of display maps, and hence of Σ?

$^+$§5.1.8

Proposition Pullback against a display map, considered as a functor between relative slices, has a right adjoint.

Proof Given $X \in \mathsf{P}A$, $B = \Sigma X$, $\alpha : B \to A$, $Y \in \mathsf{P}B$ and $a \in A$ we want $\Pi\alpha Ya$ to be the set of maps from Xa to Y over B. We have $\Pi\alpha = \lambda Yav.\mathsf{S}(\mathsf{P}BY)[\mathsf{P}(Xa)v]$ The adjunctive correspondence between $f : \mathsf{P}\alpha U \to Y$ and $g : U \to \Pi\alpha Y$ for $U \in \mathsf{P}A$ is given by $fbu = gaub \in Yb$, where $\alpha b = a \in A$, $b \in Xa \subset B$ and $u \in Ua$. The unit $\eta : U \to \Pi\alpha(\mathsf{P}\alpha U) \cong U_A^X$ and counit $\epsilon : \mathsf{P}\alpha(\Pi\alpha Y) \to B$ are given by reducing $\lambda aub.u$ and $\lambda bv.vb$ to appropriate types. $\square$

$^+$**§5.1.9** Now we shall look at the effect of the foregoing construction in the case where Λ is an unspecified poset model of the λ-calculus. We deliberately avoid being precise about which model we are using or which maps arise: the purpose of this discussion is to motivate what we may do in the *category* of domains. We began with the assumption that we represent a type which is the result of a computation by a function into the universe Λ taking type (*i.e.* idempotent) values. Now we examine the effect of that in the category, *i.e.* to what the order relation and directed sups correspond.

Lemma Let $A, B \in \mathbf{Retr}(\Lambda)$ where Λ is an β-model in **IPO**. Suppose $A \le B$ as elements of Λ. Then

(a) $A\,;B$, $B\,;A$, $A\,;B\,;A$ and $B\,;A\,;B$ are also idempotents in Λ, with the order indicated in the figure.

(b) there is an adjunction between A and B whose image in A is the closure $A\,;B\,;A$ and in B the coclosure $B\,;A\,;B$, these being isomorphic objects of $\mathbf{Retr}(\Lambda)$.

(c) if A and B commute then the diamond collapses.

$$
\begin{tikzcd}
 & B & \\
 & B\,;A\,;B \arrow[u] & \\
A\,;B \arrow[ur] & & B\,;A \arrow[ul] \\
 & A\,;B\,;A \arrow[ul] \arrow[ur] & \\
 & A \arrow[u] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
B \arrow[d, shift left=2, "B\,;A"] \\
A \arrow[u, shift left=2, "A\,;B"] \arrow[u, phantom, "\dashv"]
\end{tikzcd}
$$

(a) (b)

Figure 5.1.9: Order on idempotents in a domain λ-model

(d) if they are both coclosures then $A = B\,;A\,;B$ is a coclosure of B

(e) if they are both closures then $B = A\,;B\,;A$ is a closure of A.

Proof

[a] $A\,;B = A\,;A\,;A\,;B \le A\,;B\,;A\,;B \le A\,;B\,;B\,;B = A\,;B$, so $(A\,;B)^2 = A\,;B$. Also $A = A^3 \le A\,;B\,;A \le A\,;B^2 = A\,;B = A^2\,;B \le B\,;A\,;B \le B^3 = B$. Similarly with $B\,;A$.

[b] $(A\,;B)\,;(B\,;A) = A\,;B\,;A \ge A = 1_A$ and similarly the other way.

[c-e] obvious. $\square$

Question Suppose $F : \Lambda \to \Lambda$ preserves idempotence. Does it then preserve the above diagram?

$^{+}$§**5.1.10** An A-indexed domain therefore gives rise to a functor $X : A \to (\mathcal{C}^{hm})^{op}$, where $\mathcal{C}^{hm}$ is the category of domains and continuous functions with left adjoint. We saw from the algebraic and logical examples of indexed categories that the morphisms of the category which arises here are the homomorphisms of the structure. This gives one motivation of the definition of a *homomorphism of domains* as a continuous function with left adjoint; a surjective homomorphism is known as a *projection* and its left adjoint an *embedding*. Since the left adjoint to a homomorphism arises in this way from an instance of the order relation, we call it a *comparison* and write $(\mathcal{C}^{hm})^{op} = \mathcal{C}^{cp}$.

Lemma

(a) Let $f : X \to Y$ in $\mathbf{IPO}^{hm}$. Then there is a coclosure u on X and a closure v on Y with isomorphic images, such that f factorises as the projection $u : X \to I$ followed by an isomorphism followed by the inclusion (injective homomorphism, not embedding) $v : I \hookrightarrow Y$.

(b) Let c be a coclosure on an object $X \in \mathcal{C} \simeq \mathbf{Retr}(\Lambda)$. Then there are retracts $s \le r$ in Λ such that $\|r\| \cong X$ and $s \le r$ gives rise to the coclosure c.

(c) Likewise for a closure, $r \le s$.

Proof

[a] Let $g \dashv f$ and put $u = f\,;g \le 1_X$, $v = g\,;f \ge 1_Y$. Then $f = f\,;g\,;f = f\,;v = u\,;f$.

[b] Choose r arbitrarily with $\|r\| \cong X$, and put $s = r\,;c\,;r$, *i.e.* the extension of c from X to Λ. Then $s \leq r$ and $c = s \restriction X$ as required.

[c] Likewise. □

This lemma shows that we cannot restrict the category $\mathcal{C}^{cp}$ (*i.e.* the codomain of the functor $A \to \mathcal{C}^{cp}$ which arises from an indexed domain) further, at least so long as we insist that it contain all isomorphisms from $\mathcal{C}$. Of course if we restrict attention in advance to either closures or coclosures of a saturated domain (as is commonly done), we obtain a corresponding restriction on the category of comparisons which arise. However I believe I have seen no argument justifying such a restriction, apart from the observation that many authors seem unaware that the theory about embedding-projection pairs (*vis à vis* exponentials, recursive domain equations, *etc.*) generalises to homomorphisms which are not necessarily onto.

It is unlikely that every homomorphism will actually arise from an instance of the order relation in an arbitrary domain model: it may not be possible to choose appropriate representing idempotents for the given domains. However it is in fact possible to build a new model with the same (up to equivalence) category of retracts in which every homomorphism does occur.

$^+$§**5.1.11** We have only considered the order relation on Λ and deduced that we have a *functor* $X : A \to \mathcal{C}^{cp}$. However $X : A \to \mathsf{Idem}(\Lambda)$ is continuous (preserves directed sup) and so we have to see the effect of that.

Lemma Let Λ be a domain model and $\{A_i \in \Lambda : i \in I\}$ a directed set of idempotents. Then $A = \bigvee^{\uparrow} A_i$ is idempotent and is the bilimit of the A_i with comparisons given in the manner shown before.

Proof A is idempotent by continuity of composition. We have coclosures $A \to A_i \to A$ given by $A\,; A_i\,; A$, and the directed sup of these is clearly $A^3 = A = 1_A$. □

Note This argument only shows $A \hookrightarrow \mathsf{bilim}\, A_i$; a full proof will be found in *Homomorphisms, Bilimits and Saturated Domains.*

$^+$§**5.1.12** We have now justified the definition of a *continuous type-dependence* or *indexed domain* as a Scott-continuous functor into $\mathcal{C}^{cp}$, the category of domains and *comparisons*.

Examples

(a) $\mathsf{Idem}(\Lambda) \to \mathbf{Retr}(\Lambda)$, where $\mathsf{Idem}(\Lambda) \subset \Lambda$ is the poset (not category) of idempotents (although this is not necessarily a domain).

(b) $\downarrow : A \to \mathbf{bcCont}$ for $A \in \mathbf{bcCont}$ (the homomorphisms are of the form $- \curlywedge a$).

(c) $\uparrow : A \to \mathbf{ContLat}$ for $A \in \mathbf{ContLat}$ (the comparisons being $- \curlyvee a$).

(d) $\Phi : (\mathcal{C}^{cp})^n \to \mathcal{C}^{cp}$ where $\Phi(X_1, ..., X_n)$ is any expression in type variables $X_1, X_2, ..., X_n$ and constructors $\times$, $\to$, *etc.* Again $(\mathcal{C}^{cp})^n$ is not a domain.

$^+$§**5.1.13** As a footnote to this motivation, consider an indexed domain as a "domain about which we have partial information". Thus at a particular value $a \in A$ of the parameter we know $x_a \in X_a$ about the data. If we have obtained a by improvement from b with $\alpha : b \leq a$, then we take with this at least a certain degree of knowledge, say $\Sigma\alpha x_b$ about x. Conversely if we discard information, reducing from a to b, then we can know at most $\mathsf{P}\alpha x_a$. Performing these two processes, which as usual we assume to be continuous, we deduce $x_b \leq \mathsf{P}\alpha(\Sigma\alpha x_b)$ and

$\Sigma\alpha(\mathtt{P}\alpha x_a) \le x_a$. Hence $\Sigma\alpha \dashv \mathtt{P}\alpha$. Further, if we can approximate a as $\bigvee^{\uparrow} a_i$, then we should be able to recover x_a as $\bigvee^{\uparrow} \Sigma\alpha_i(\mathtt{P}\alpha_i x_a)$. Hence $\mathtt{P} : A \to \mathcal{C}^{cp}$.

Since we have adjoints to substitution in an indexed category, it is natural to ask whether we should require the Beck condition to hold. The answer to the appears to be "no": we carry through the theory without it, getting arbitrary (*i.e.* non-Beck) indexations, and it just seems that there are some circumstances (*viz.* logic) where we need the Beck conditions and others (both here and in examples in Universal Algebra) where we dont.

In what follows we have tended to use $\mathtt{P}$ for indexations and Σ for their left adjoints, despite the fact that the use of this letter misleadingly suggests the Beck condition and logic. This is undoubtedly a mistake, and a revision of this notation would be appropriate. Where the indexation has another name, say X or even $\Pi\alpha Y$, we still use Σ, but subscripted appropriately, *e.g.* Σ_X and $\Sigma_{\Pi\alpha Y}$.

5.2 Indexed Categories of Domains

$^+$§**5.2.1** The previous section culminated in the definition of a continuous type-dependence as a *continuous* functor from a domain to the category of domains and *comparisons.* We shall see that for each of the many categories of domains which we introduced in Chapter IV (large or small, continuous or algebraic and — more strikingly — bifinite or *any* category of boundedly-complete posets) the theory is very rich and beautiful.

Specifically,

(a) there is a direct correspondence between indexations $\mathtt{P} : A \to \mathcal{C}^{cp}$ and fibrations $p : X \to A$, *i.e.* $X \in \mathcal{C}$ iff $A \in \mathcal{C}$ and $\forall a \in A.\mathtt{P}a \in \mathcal{C}$.

(b) this class of display maps yields a relatively cartesian closed category, *i.e.* we have domain-indexed sums and products of domains (even in the small case).

(c) we have an indexed form of domain theory, *i.e.* indexed bilimits and fixpoints.

On the face of it, however, the conditions of the definition appear to be very strong, and (in comparison with the theory for categories) it may appear that we have not motivated either of the requirements that the substitution maps have left adjoints or that the functor be continuous (*i.e.* send directed sups to bilimits). It is therefore appropriate to consider weakened forms of the definition, both from the indexed and the fibred points of view; we find that the theory fails to be at all satisfactory.

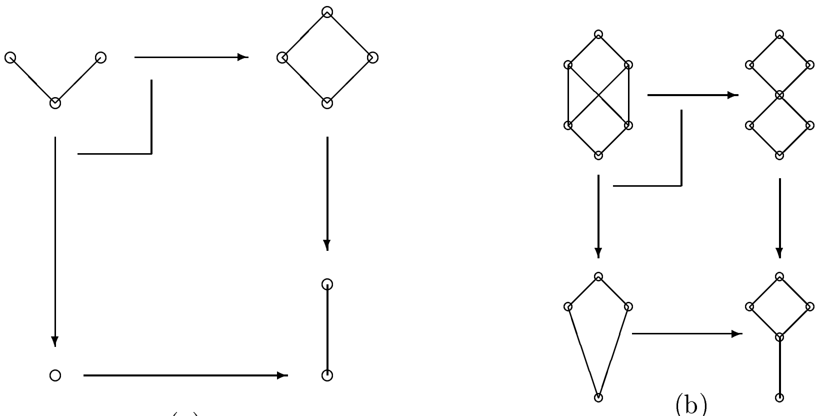
$^+$§**5.2.2** The most obvious candidate for a class of display maps for **IPO** is the projections, as strongly hinted by Lemma 2.1.8. However, contrary to the claims of [Taylor 1986], they do not give a class of displays for **bcCont**.

Examples Pullback may destroy top or bounded completeness, as in the figure.

The fallacy in the alleged proof of the first proposition on page 463 of [Taylor 1986] lies in the last line, where it is claimed that we may pull back an idempotent on W to one on $W \times_{A^\top} B^\top$.

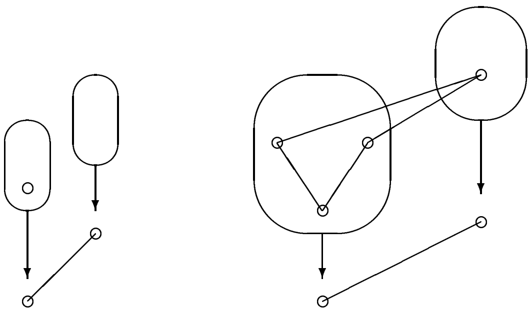
$^+$§**5.2.3** When we spoke of the "fibred" form of a continuous type-dependence in the previous paragraph, we meant the fibration constructed from the indexation in the manner of Chapter IV, where as usual we regard a domain as a poset and hence as a category.

In most cases when studying posets we have no need to give a name to the (unique) instance of the comparison between any two elements, and for this reason lattice theory tends to be substantially simpler than category theory. However in our case we *have* to regard the base object A as a category and give names to the comparisons in it in order to name the substitution functors which arise. In order to avoid repetitious data, we shall adopt the convention that a, a', a_i and b



(a) (b)

Figure 5.2.2: Pullback may destroy top or bounded completeness



(a) (b)

Figure 5.2.3: Projections and fibrations

are elements of A, and that $\alpha : b \le a$, $\alpha_{ij} : a_i \le a_j$, $\alpha_i : a_i \le a$ are instances of the comparison. Because we are forced to use the general notation from the start, much of the theory carries over immediately to categories rather than posets.

Lemma Let $p : X \to A$ be a fibration in which each fibre is inhabited. Then

(a) p has a left adjoint iff each fibre has an initial object (least element, $\perp_a$), and this is $a \mapsto \perp_a$.

(b) (Ignoring questions of choice) the substitution functors have left adjoints iff $p : X^{op} \to A^{op}$ is also a fibration.

Proof

[a,$\Rightarrow$] Let $i \dashv p$ and $\perp_a = ia$ for $a \in A$; since p is surjective, $\perp_a \in \mathtt{P}a$. Then $f : \perp_a \to X$ correspond bijectively to $pf : a \to px$ and in particular there is a unique $f : \perp_a \to X$ over $1_{\mathtt{P}a}$.

[a,$\Leftarrow$] Put $ia = \perp_a$. Then for $\beta : a \to b$, $x \in \mathtt{P}b$, by definition of $\perp_a$ there's a unique map $\perp_a \to \mathtt{P}\beta x$. Hence $ia \to x$ correspond bijectively to $\beta : a \to b = px$.

[b] This is proposition 4.1.12. $\square$

If, however, any fibre is empty or fails to have bottom, a fibration will not be a projection. Conversely the coclosure on the four-point lattice which fixes top and bottom gives a projection which is not a fibration.

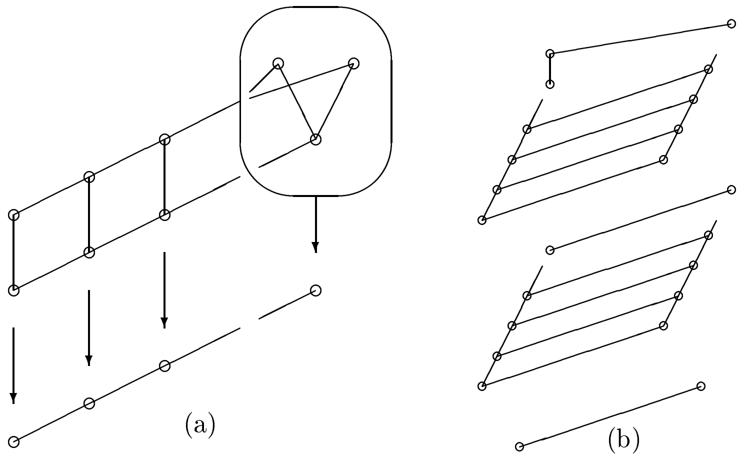


Figure 5.2.4: Continuous fibration counterexamples

$^+$§**5.2.4** For technical reasons within the constructions, it is worthwhile introducing a weakened form of the definition in which we drop the requirement for adjoints to substitution.

Definition A *continuous indexation of domains* is a functor $\mathtt{P} : A \to \mathbf{IPO}^{op}$ which takes directed sups to cofiltered limits. Thus for any directed set $(a_i : i \in I) \subset A$, say with $a = \bigvee^{\uparrow} a_i$, we have a cofiltered diagram $\mathtt{P}\alpha_{ij} : \mathtt{P}a_i \to \mathtt{P}a_j$ and a *limiting* cone $\mathtt{P}\alpha_i : \mathtt{P}a \to \mathtt{P}a_j$. Hence elements of $\mathtt{P}a$ are given by *compatible families*, $(x_i \in \mathtt{P}a_i : i \in I)$, where $x_j = \mathtt{P}\alpha_{ij}x_i$.

Definition A *continuous fibration of domains* is a map $p : X \to A$ in **IPO** such that

(i) p is a fibration in the sense of Chapter IV,

(ii) each fibre has a least element (equivalently by lemma 5.2.3a, p is a projection in **IPO**),

(iii) the substitution maps $\mathtt{P}\alpha$ are continuous,

(iv) given $px = a = \bigvee^{\uparrow} a_i$, then $x = \bigvee^{\uparrow} \mathtt{P}\alpha_i x$, where $\alpha_i : a_i \leq a$,

and

(v) given a compatible family $x_i \in \mathtt{P}a_i$ (for the substitution maps) over $a = \bigvee^{\uparrow} a_i$, then $x_i = \mathtt{P}\alpha_i(\bigvee^{\uparrow} x_j)$.

If also

(vi) $p : X^{op} \to A^{op}$ is a fibration

then we say p is a *fibred type*.

For fibred domain theory the first three conditions are obviously natural. (iv) and (v) express in detail the continuity of the indexation, *i.e.* that directed sups go to cofiltered limits, and one might question them. However these conditions are used very frequently in the constructions which follow, and occur in much the same rôle as continuity of substitution. Questions of the usefulness of domain theory without continuity I shall leave to the reader's imagination.

Examples

(a) Given a discontinuous functor $A \to \mathbf{IPO}^{op}$, the corresponding fibration $X \to A$ need not have $X \in \mathbf{IPO}$.

(b) A composite of fibrations between ipos with continuous substitutions need not have continuous substitutions.

$^+$§**5.2.5** In the theory which follows, we shall make heavy use of the correspondence between the indexed and fibred presentations, so in particular we have to show that the definitions of the previous paragraph are equivalent. First we construct fibrations from indexations.

Lemma Let $A \in \mathbf{IPO}$ and $\mathtt{P} : A \to \mathbf{IPO}^{op}$ be a continuous indexation (sending directed sup to cofiltered limit), and $p : X \to A$ the corresponding fibration. Then X is an ipo and p a projection.

Proof There are no nontrivial parallel pairs of maps in the fibres or horizontally, so X is a poset; p has a left adjoint by lemma 5.2.3a. We have to show that X has and p preserves directed sups.

Let $(x_i : i \in I) \subset X$ be directed with $a_i = px_i$ and $a = \bigvee^{\uparrow} a_i$. For $i \le j$ let $\alpha_{ij} : a_i \le a_j$ and $\alpha_i : a_i \le a$. By hypothesis the substitution maps over α_i form a limiting cone over the cofiltered diagram given by the α_{ij}. For $i \le j$, $x_i \le y_{ij} = \mathtt{P}\alpha_{ij} x_j \le x_j$. Hence for each i, $\{y_{ij} : i \le j\}$ is a directed subset of $\mathtt{P}a_i$, so put y_i for its sup.

Then

$$\begin{aligned}
\mathtt{P}\alpha_{ij} y_j &= \mathtt{P}\alpha_{ij} \bigvee\nolimits^{\uparrow}(\mathtt{P}\alpha_{jk} x_k : j \le k) && \text{definition}\\
&= \bigvee\nolimits^{\uparrow}(\mathtt{P}\alpha_{ij}(\mathtt{P}\alpha_{jk} x_k) : j \le k) && \text{continuity}\\
&= \bigvee\nolimits^{\uparrow}(\mathtt{P}\alpha_{ik} x_k : j \le k) && \text{functoriality}\\
&= \bigvee\nolimits^{\uparrow}(\mathtt{P}\alpha_{ik} x_k : i \le k) && \text{cofinality}\\
&= y_i && \text{definition}
\end{aligned}$$

so $y = (y_i : i \in I)$ is a compatible family for the diagram and hence an element of $\mathtt{P}a$.

If $\forall i. x_i \le z \in \mathtt{P}b$ with $\beta_i : a_i \le b$ and $\beta : a \le b$ then

$$y_i \quad = \quad \bigvee\nolimits^{\uparrow}\{\mathtt{P}\alpha_{ij} x_j : i \le j\} \quad \le \quad \bigvee\nolimits^{\uparrow}\{\mathtt{P}\alpha_{ij}(\mathtt{P}\beta_j z) : i \le j\} \quad = \quad \mathtt{P}\beta_i z \quad \le \quad z$$

so $y \le z$. Thus $y = \bigvee^{\uparrow} x_i \in X$ and $py = a$. Hence X has an p preserves $\bigvee^{\uparrow}$. □

Notice that compatible families correspond to directed sups. This result does not require the substitutions to be comparisons, only that directed sups go to cofiltered limits.

Proposition Proposition 4.2.4 specialises to equivalences between

(a) continuous indexations and continuous fibrations of ipos

(b) continuous type-dependences and fibred types.

Proof

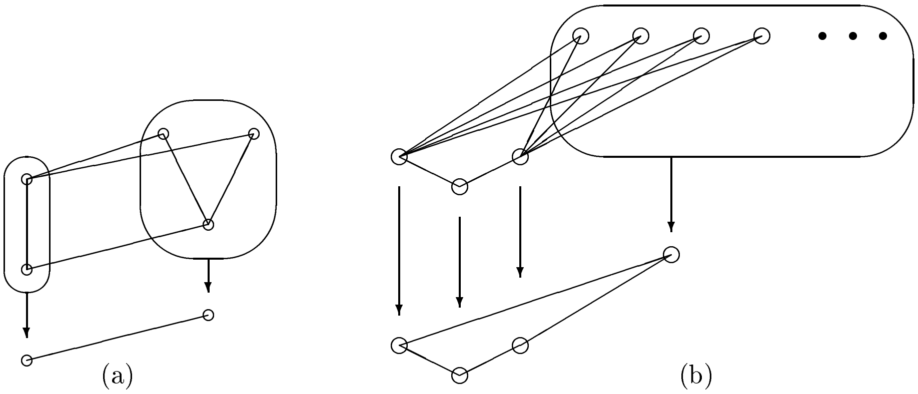
[a,⇒] By the lemma.

[a,⇐] We have only to check that the fibres are ipos; but these are given by pullback of the fibration (which is a projection) against the corresponding global element (map $1 \to A$) of the base.

[b] lemma 5.2.3b □

$^+$§**5.2.6** We have established the equivalence between the indexed and fibred approaches for continuous indexations and continuous type-dependences of ipos, and as we have already remarked this will be seen to apply to continuous type-dependences of *any* of the categories of domains which we have studied. However for continuous indexations of categories other than **IPO** and weaker attempted classes of displays for even **IPO**, the equivalence breaks down. Curiously, it does so in opposite ways for the cases of **bcCont** and **AlgPos**, and in both ways for $\mathbf{BiPos}_{\mathfrak{f}}$.

Lemma Let $\mathtt{P} : A \to \mathbf{IPO}^{op}$ be a continuous indexation with fibration $p : X \to A$.



(a) (b)

(a) If A and each $\mathtt{P}a$ are algebraic then so is X.

(b) If X is boundedly complete then so are A and $\mathtt{P}a$.

Proof

[a] We shall prove essentially this in §5.3.2.

[b] It is easy to show that p preserves mubs. □

Examples

(a) The display of a continuous indexation of boundedly complete posets need not be boundedly complete.

(b) Likewise for bifinite posets (the base is a four-point lattice and only the top fibre is nontrivial).

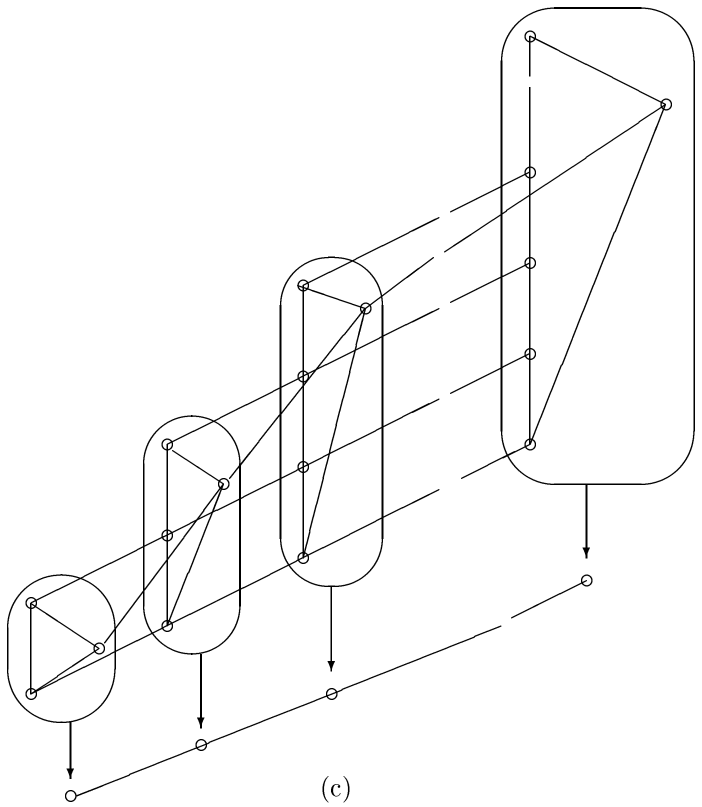
(c) Conversely a continuous fibration of algebraic posets need not have algebraic fibres. The "odd" point in the infinite fibre is compact neither in the fibre nor in the display, but in the display it is approximated. This of course also shows that **AlgPos** is not closed under arbitrary cofiltered limits.

(d) Likewise for bifinite posets. In this example the finite points of the display are the points of the finite fibres, and any finite number of fibres yields a mub-closed set; however restricting to the infinite fibre, the two almost-minimal points become finite but have a Cantor set of mubs.

$^{+}$§**5.2.7** Since domain theory, unlike category theory, is infinitary, we still have some groundwork to do before showing that we have classes of display maps even for **IPO**. First we apply Proposition 4.2.18 about limits of fibrations to the domain case.

Proposition Let $p_i : X_i \to A_i$ be a compatible family of (a) projections, (b) fibrations, (c) continuous fibrations or (d) fibred types, for the filtered diagrams $X = \lim X_i$, $A = \lim A_i$. This means (for b–d) that we ask for the diagram

$$\begin{tikzcd} X_j \arrow[r, "f_{ji}"] \arrow[d, "p_j"] & X_i \arrow[d, "p_i"'] \\ A_j \arrow[r, "g_{ji}"] & A_i \end{tikzcd}$$



(c)

Figure 5.2.6: Continuous indexation counterexamples

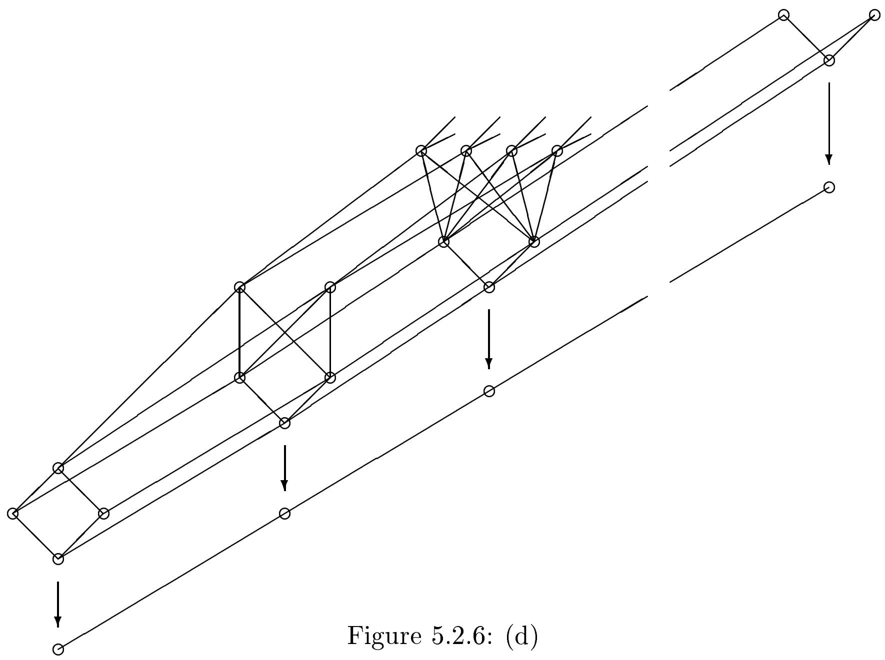


Figure 5.2.6: (d)

to be an indexed functor. Then the mediating map $p : X \to A$ is a projection, *etc.*

Proof First, p is continuous.

[a] Let $\iota_i \dashv p_i$. For $a \in A$ define $(\iota a)_i = \bigvee^{\uparrow}_{i \le j} f_{ji}(\iota_j a_j)$. We may show that this is a compatible family in the same way as in lemma 5.2.5 and also that $\iota \dashv p$. However it is crucial to use the fact that (a_i) and (x_i) are compatible families.

[b] Proposition 4.2.18.

[c] Continuity is preserved because we may interchange limits over $\mathtt{P}_i a_{ij}$.

[d] Use lemma 5.2.3b. □

${}^{+}$§5.2.8

Lemma A composite of continuous fibrations is again a continuous fibration.

Proof Let $\mathtt{P} : A \to \mathbf{IPO}^{op}$ and $\mathtt{Q} : B \to \mathbf{IPO}^{op}$ be continuous indexations with corresponding fibrations $p : B \to A$ and $q : C \to B$ respectively, and $r = q\,;p$ with corresponding indexation the *functor* $\mathtt{R} : A \to \mathbf{Cat}^{op}$; we have to show that each $\mathtt{R}a$ is an ipo that the substitution maps are continuous and that directed sups map to limits.

p, q and r are projections in **IPO** by proposition 5.2.5a. For $a \in A$, $\mathtt{R}a$ is an ipo because it is $r^{-1}(a) \subset C$; alternatively, as remarked in §4.2.15, it is the fibration of $\mathtt{Q} \restriction \mathtt{P}a$.

Now let $\alpha : a' \le a$ in A; we have to show that $\mathtt{R}\alpha : \mathtt{R}a \to \mathtt{R}a'$ is continuous. This is the cartesian functor (§4.2.6) corresponding to the indexed form $\mathtt{Q}\beta : \mathtt{Q}b \to \mathtt{Q}(\mathtt{P}\alpha b)$ for $b \in \mathtt{P}a$ over a change of base (§4.2.13). To show continuity of $\mathtt{R}\alpha$, let $\{c_i : i \in I\} \subset \mathtt{R}a$ be directed, with $c_i \in \mathtt{Q}b_i$. Put $c = \bigvee^{\uparrow} c_i$ and $b = \bigvee^{\uparrow} b_i$. Then since $\mathtt{Q} : B \to \mathbf{IPO}^{op}$ is continuous, $c \in \mathtt{Q}b \cong \lim(\mathtt{Q}b_i)$. With $\beta_i : \mathtt{P}\alpha b_i \le b_i$, $\beta : \mathtt{P}\alpha b \le b$ the following diagram commutes:

$$\begin{tikzcd}
\mathtt{Q}(\bigvee^{\uparrow} b_i) \cong \lim \mathtt{Q}b_i \arrow[r] \arrow[d, "\mathtt{Q}\beta"'] & \mathtt{Q}b_i \arrow[d, "\mathtt{Q}\beta_i"'] \\
\mathtt{Q}(\mathtt{P}\alpha(\bigvee^{\uparrow} b_i)) \cong \mathtt{Q}(\bigvee^{\uparrow}(\mathtt{P}\alpha b_i)) \cong \lim \mathtt{Q}(\mathtt{P}\alpha b_i) \arrow[r] & \mathtt{Q}(\mathtt{P}\alpha b_i)
\end{tikzcd}$$

(using continuity of $\mathtt{P}\alpha$). Then $\mathtt{R}\alpha c = \mathtt{Q}\beta c = (\mathtt{Q}\beta_i c_i) = (\mathtt{R}\alpha c_i)$ so $\mathtt{R}\alpha$ is continuous.

Now we have to show that $\mathtt{R} : A \to \mathbf{IPO}^{op}$ is continuous. Let $\{a_i : i \in I\} \subset A$ be directed and $a = \bigvee^{\uparrow} a_i$. We have a compatible family of continuous fibrations $\mathtt{R}a_i \to \mathtt{P}a_i$, so the mediating map $\lim(\mathtt{R}a_i) \to \lim(\mathtt{P}a_i) \cong \mathtt{P}a$ is a continuous fibration by proposition 5.2.7c. The fibre of this over $b \in \mathtt{P}a$ is $\lim(\mathtt{Q}(\mathtt{P}\alpha_i b))$ by this construction, and this is isomorphic to $\mathtt{Q}b$ since $b = \bigvee^{\uparrow}(\mathtt{P}\alpha_i b)$ and $\mathtt{Q}$ is continuous; but this is the fibre of $\mathtt{R}a$ over b. □

Proposition The following give classes of display maps for **IPO**, each class being contained in the previous one:

(a) projections, *i.e.* continuous surjections with left adjoint,

(b) fibrations (in the categorical sense) which are also projections,

(c) continuous fibrations, and

(d) fibred types.

Proof Nesting is obvious. By lemma 4.2.15 a composite of fibrations is a fibration, as is any terminal projection; the latter correspond to type-dependences on 1 and so are fibred types and hence in all four classes.

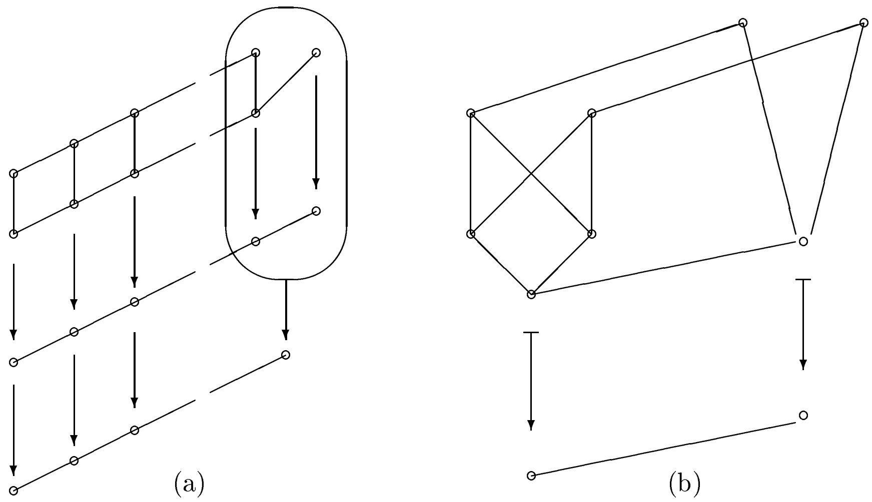


Figure 5.2.9: Indexed product counterexamples

[a] proposition 2.1.8

[b] the classes of fibrations and of projections are each closed under composition and pullback, and hence so is their intersection.

[c] the lemma gives closure under composition; for pullback we use the equivalence with indexations (proposition 5.2.5a), when pullback becomes precomposition.

[d] the composite of the fibrations $p : B^{op} \to A^{op}$ and $q : C^{op} \to B^{op}$ is a fibration; for pullback we use proposition 5.2.5b. $\square$

$^{+}$§**5.2.9** There appear still to be three competitors besides the definition given at the end of the previous section for continuous type-dependences (fibred types). However they fail to have indexed products (exponentials);

Examples

(a) Contrary to the claim of [Taylor 1986] and the previous draft of this dissertation, **IPO** is not cartesian closed relative to projections. In this example, $\Pi\alpha Y$ does not have $\bigvee^{\!\uparrow}$.

(b) A finitary example covers fibrations and continuous fibrations. Writing X over A for the fibration below, X_A^X is not a fibration. For the unique nontrivial base-morphism has no cartesian lifting at a certain point. This is the point in the upper fibre of X_A^X which is the identity on the upper fibre of X. The identity on the lower fibre of X and the involution of this fibre interchanging the lower two points provide points of the lower fibre of X_A^X which are both maximal below the upper identity. $\square$

5.3 Categories of Fibred Types

$^{+}$§**5.3.1** From now on we shall discard all but the chosen definition of fibred types. As remarked, there is a remarkably exact correspondence between the indexed and fibred approaches in this case, and my personal belief is that this is the strongest evidence that we are on the right track. This correspondence has to be proved separately for each of our categories of domains, and we begin with two proofs for $\mathbf{BiPos}_{\mathrm{f}}$ — I have unfortunately not found the right trick to deal with **ContDom**. We have to take bilimit expressions for A and $\mathtt{P}a$ and transform them into such expressions for X, and some work is needed to achieve the necessary uniformity in the expressions.

As a notational abbreviation, $\mathtt{P} : A \to \mathcal{C}^{cp}$ is a continuous type-dependence with fibration $p : X \to A$, where $\mathcal{C}$ is a category of domains and it is intended that $A, X \in \mathcal{C}$ (though the object is usually to *prove* the latter).

$^{+}$§**5.3.2** We shall prove the $\mathbf{BiPos}_{\mathrm{f}}$ case using finite elements and mub-closed sets. The following lemma classifies finite elements in the fibration.

Lemma $x \in X$ is finite iff $a = px \in A_{fp}$ and $x \in (\mathtt{P}a)_{fp}$.

Proof

[$\Rightarrow, A$] Suppose $a \leq b = \bigvee^{\uparrow} b_i$ in A, say $\beta_i : b_i \leq b$, $\gamma : a \leq b$. Put $y = \Sigma\gamma x \in \mathtt{P}b$ and $y_i = \mathtt{P}\beta_i y \in \mathtt{P}b$, so $x \leq y = \bigvee^{\uparrow} y_i$. But $x \in X_{fp}$ so $x \leq y_i$ for some i, so $a \leq b_i$.

[$\Rightarrow, \mathtt{P}a$] Suppose $x \leq y = \bigvee^{\uparrow} y_i$ with $y_i \in \mathtt{P}a \subset X$. But $x \in X_{fp}$, so $x \leq y_i$ for some i.

[$\Leftarrow$] Suppose $x \leq y = \bigvee^{\uparrow} y_i$, $y_i \in \mathtt{P}b_i$, $y \in \mathtt{P}b$, $b = \bigvee^{\uparrow} b_i$, so $a \leq \bigvee^{\uparrow} b_i$. But $a \in A_{fp}$ so $a \leq b_i$ for some i. This holds for all later i, so w.l.o.g. $\forall i.a \leq b_i$. Let $\gamma_i : a \leq b_i$, $\gamma : a \leq b$. Put $z_i = \mathtt{P}\gamma_i y_i \in \mathtt{P}a$, so $x \leq z = \bigvee^{\uparrow} z_i = \mathtt{P}\gamma_i y$; but $x \in (\mathtt{P}a)_{fp}$ so $x \leq z_i \leq y_i$ for some i. □

Proposition $X \in \mathbf{AlgPos}$ iff $A \in \mathbf{AlgPos}$ and $\forall a \in A.\mathtt{P}a \in \mathbf{AlgPos}$.

Proof

[$\Rightarrow, A$] By the lemma, $A_{fp} = p(X_{fp})$. Let $a \in A$. If $x \in X_{fp}$ and $x \leq \bot_a$ then $px \in A_{fp}$ and $px \leq a$; moreover $a = \bigvee^{\uparrow}(px : x \in X_{fp} \cap {\downarrow}\bot_a)$ so A is algebraic.

[$\Rightarrow, \mathtt{P}a$] Since **AlgPos** is closed under bilimits, A is algebraic and $\mathtt{P}$ continuous, it suffices to check that $\mathtt{P}a \in \mathbf{AlgPos}$ for $a \in A_{fp}$. By the lemma, $(\mathtt{P}a)_{fp} = \mathtt{P}a \cap X_{fp}$. Let $x \in \mathtt{P}a$. If $y \in X_{fp} \cap {\downarrow}x \cap \mathtt{P}b$ with $\alpha : b \leq a$ then $y \leq \Sigma\alpha y \in X_{fp} \cap {\downarrow}x \cap \mathtt{P}a$, so $x = \bigvee^{\uparrow}[(\mathtt{P}a)_{fp} \cap {\downarrow}x]$ and $\mathtt{P}a$ is algebraic.

[$\Leftarrow$] For $a \in A$, $a = \bigvee^{\uparrow} A_{fp} \cap {\downarrow}a$, so by continuity $\mathtt{P}a \cong \mathsf{bilim}(\mathtt{P}b : b \leq a \wedge b \in A_{fp})$. This means that $x \in \mathtt{P}a \subset X$ is approximated by $\mathtt{P}\alpha x \in \mathtt{P}b \subset X$ for such $\alpha : b \leq a$. These are approximated in turn by the finite elements of $\mathtt{P}b$ (which are also finite in X) below them. □

$^{+}$§**5.3.3**

Proposition $X \in \mathbf{BiPos}_{\mathrm{f}}$ iff $A \in \mathbf{BiPos}_{\mathrm{f}}$ and $\forall a \in A.\mathtt{P}a \in \mathbf{BiPos}_{\mathrm{f}}$

Proof We have only to check mub-closed sets.

[$\Rightarrow, A$] Let $S \subset A_{fp}$ be finite and M the mub-closure of $\{\bot_s : s \in S\} \subset X_{fp}$. Then pM is mub-closed. For let $a \in A$ and x be the projection of $\bot_a$ into M, then $px \in pM \cap {\downarrow}a$; if also $b \in pM \cap {\downarrow}a$, $\bot_b \leq m \leq a$ for some $m \in M \cap \mathtt{P}b$, so $\bot_b \leq x$ and $b \leq px$.

[$\Rightarrow, \mathtt{P}a$] Let $S \subset (\mathtt{P}a)_{fp}$ be a finite set and M be the mub-closure of $S \cup \{\bot_a\}$ in X. I claim $M' = M \cap \mathtt{P}a$ is mub-closed in $\mathtt{P}a$. For let $x \in \mathtt{P}a$ and y be its reduction to M with $\alpha : b = py \leq a$. Now $\bot_a \in S \subset M$ so $\bot_a \leq y$ and $a \leq b$; hence $y \in \mathtt{P}a \cap M = M'$.

[$\Leftarrow$] Let $S \subset X_{fp}$. Put $T = pS \subset A_{fp}$ and N for its mub-closure in A. This is a finite poset, so for $a \in N$ we may define M_a inductively as the mub-closure of

$$(S \cap \mathtt{P}a) \cup \{\Sigma\alpha m : m \in M_b, \alpha : b < a\}$$

Put $M = \bigcup_{a \in A} M_a$; this is a finite set of finite elements, and I claim it is mub-closed. Let $x \in \mathtt{P}a \subset X$. Put b for the reduction of a to N with $\alpha : b \leq a$. Then $M \cap {\downarrow} x \leq \mathtt{P}\alpha x$. Let y be the reduction of $\mathtt{P}\alpha x$ to M_b. Now suppose $z \in M$ with $z \leq x$, say $\beta : c = pz \leq b$. Then $z \leq \Sigma\beta z \leq \mathtt{P}\alpha x \leq x$. But $\Sigma\beta z \in M_b$ by construction, so $\Sigma\beta z \leq y$. Thus y is the reduction of x to M. $\square$

The crucial point of this proof is that $\Sigma\alpha$ restrict to $M_b \to M_a$.

$^+$§**5.3.4** Having seen a "concrete" version of this proof in terms of finite elements, we are in a position to give a categorical version. This depends on $\mathsf{Cocl}(X)$ being a continuous lattice (proposition 2.4.10), so that we may express X as a bilimit of Y for which the coclosure $c : X \to Y \to X$ has $c \ll 1_X$ (we shall abuse notation by writing $Y \ll X$ for this). The delicacy of this construction (or rather the crudeness with which it has been implemented) necessitates restricting in the first instance to finite A. If I had the appropriate trick for retracts, this result would form a case in the proof by induction on the complexity of the domain (§2.5.2) that **ContDom** is closed under the fibration construction.

Note "Proposition" 2.4.10 is not valid in general.

Lemma Let $\sigma : X \to Y$ and $X' \ll X$ in $\mathbf{ContDom}^{cp}$. Then $\sigma X' \ll Y$, *i.e.* there's a $Y' \ll Y$ through which $\sigma \restriction X'$ factorises.

Proof "$X' \ll X$" is an instance of $\ll$ in $\mathsf{Cocl}(X)$, which is embedded in ${\downarrow} 1_X \subset [X \to X]$: recall that comparisons preserve and embeddings also reflect $\ll$. Write $c : X \to X$ for the coclosure corresponding to X' and $\sigma \dashv p$. Now $\sigma(c) = p;c;\sigma$ is a coclosure on Y, and $\sigma(c) \ll 1_Y$ in $\mathsf{Cocl}(Y)$. Put Y' for its splitting; then $\sigma X' = Y'$ *i.e.* $\sigma \restriction X'$ factors through Y'. $\square$

Proposition Let $A \in \mathbf{Pos}_f$ and $\mathtt{P} : A \to \mathbf{ContDom}^{cp}$ be a continuous type-dependence with fibration $p : X \to A$. Then X may be expressed as a bilimit of Y where $q : Y \to A$ are fibrations corresponding to continuous type-dependences $\mathtt{Q} : A \to \mathbf{ContDom}^{cp}$ with $\mathtt{Q}a \ll \mathtt{P}a$.

Proof Let $U_a \ll \mathtt{P}a$ be an arbitrary "reference" choice of approximants; it suffices to make $U_a \leq \mathtt{Q}a \ll \mathtt{P}a$ in order to show that X is the bilimit of such Y. Since A is finite and $\mathsf{Cocl}(\mathtt{P}a)$ is a continuous lattice, we may define

$$\mathtt{Q}a = \bigvee (\Sigma\alpha U_b : \alpha : b \leq a)$$

By the lemma, $\Sigma\alpha : \mathtt{P}b \to \mathtt{P}a$ restricts to $\mathtt{Q}b \to \mathtt{Q}a$. Also $U_a \leq \mathtt{Q}a \ll \mathtt{P}a$.

Put $\mathtt{Q}\alpha : \mathtt{Q}a \rightarrowtail \mathtt{P}a \to \mathtt{P}b \to \mathtt{Q}b$ and verify that $\Sigma\alpha \dashv \mathtt{Q}\alpha$; moreover $\mathtt{Q} : A \to \mathbf{ContDom}^{cp}$ is a functor, indeed (trivially, since A is finite) a continuous type-dependence. Let $q : Y \to A$ be the corresponding fibration; we have to show that Y is a coclosure of X; the *projection* $Y \to X$ turns out to be fibred over A, but the *embedding* is not. By construction the diagram on the left commutes:

$$\begin{tikzcd}
\mathtt{P}b \arrow[r, "\Sigma_P\alpha"] & \mathtt{P}a \\
\mathtt{Q}b \arrow[u, "\iota_b"] \arrow[r, "\Sigma_Q\alpha"] & \mathtt{Q}a \arrow[u, "\iota_a"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathtt{P}b \arrow[d, "\pi_b"] & \mathtt{P}a \arrow[l, "\mathtt{P}\alpha"'] \arrow[d, "\pi_a"'] \\
\mathtt{Q}b & \mathtt{Q}a \arrow[l, "\mathtt{Q}\alpha"']
\end{tikzcd}$$

and hence by adjointness so does that on the right. Hence the projection $\pi : \mathtt{P} \to \mathtt{Q}$ is an A-indexed functor, and we have a fibred functor $\pi : X \to Y$ over A.

Now we have to show that this has a (non-fibred) left adjoint, essentially given by ι. Thus for $y \in \mathtt{Q}a \subset Y$, $\iota y = \iota_a y$. To check monotonicity, let $y_1 \leq y_2$ over $\alpha : a_1 \leq a_2$; then $\iota_{b_1} y_1 \leq \iota_{b_2}(\mathtt{Q}\alpha y_2) \leq \mathtt{P}\alpha(\iota_{b_2} y_2) \leq \iota_{b_2} y_2$. For adjointness, let $x \in \mathtt{P}b$, $\beta : a \leq b$; then the following are equivalent:

$$\begin{array}{lll} y & \leq_\beta & \pi_b x \\ y & \leq_a & \mathtt{Q}\beta(\pi_b x)\pi_a(\mathtt{P}\beta x) \\ \iota_a y & \leq_a & \mathtt{P}\beta x \\ \iota_a y & \leq_\beta & x \end{array}$$

hence Y is a coclosure of X and $q : Y \to A$ is a fibration of the required kind. □

$^+$§**5.3.5** The restriction to finite base (A) in the previous result is not a serious one; indeed we could (at the expense of increased complexity) have included the approximation of A by $A' \ll A$ in the construction, as we did implicitly in proposition 5.3.2. However manipulating the base domain is much simpler than manipulating the fibres.

Proposition

(a) Let $A = \mathsf{bilim}\, A_i$. Then $X = \mathsf{bilim}\, X_i$, where $p_i : X_i \to A_i$ are the fibrations corresponding to $\mathtt{P}_i : A_i \rightarrowtail A \to \mathbf{IPO}^{cp}$.

(b) Let $A \triangleleft B$. Then $X \triangleleft Y$, where $Y \to B$ is the display of $B \twoheadrightarrow A \to \mathbf{IPO}^{cp}$.

Proof

[a] For $i \leq j$, $X_i \cong X_j \times_{A_j} A_i$; let the coclosure on A_j corresponding to A_i be r. Then there is a right adjoint $X_j \to X_i$ which takes $x_j \in \mathtt{P}_j a_i \subset X_j$ to $\langle \mathtt{P}_j \alpha x_j, r a_j \rangle$, where $\alpha : r a_j \leq a_j$. This is a fibred functor over $A_j \to A_i$, so p_i is a compatible family for $\mathsf{bilim}\, X_i$ over $A = \mathsf{bilim}\, A_i$. The mediating map is a fibred type with fibre $\mathsf{bilim}(\mathtt{P}_i a_i) \cong \mathtt{P}a$ over a, so $X \cong \mathsf{bilim}\, X_i$.

[b] The squares and rectangle below are pullbacks:

$$\begin{tikzcd} X \arrow[r, hook] \arrow[d, "p"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[r, two heads] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "p"'] \\ A \arrow[r, hook] & B \arrow[r, two heads] & A \end{tikzcd}$$

□

$^+$§**5.3.6**

Proposition Fibred types form a class of displays for $\mathbf{BiPos}_\mathrm{f}$ and $\mathbf{Bi}_\omega\mathbf{Pos}_\mathrm{f}$.

Proof We have to check closure under pullback and composition; terminal projections are trivially fibred types in the category. By the foregoing results we have a two-way translation between indexations and fibrations. If $p : X \to A$ is a fibred type in $\mathbf{BiPos}_\mathrm{f}$ then it corresponds to an indexation $\mathtt{P} : A \to {\mathbf{BiPos}_\mathrm{f}}^{cp}$. The pullback along $\alpha : B \to A$ of the fibration is the fibration of the precomposite with this; this composite clearly lies in $\mathbf{BiPos}_\mathrm{f}$ and hence so does the fibration. For composites, the objects are already in the category, and the question for morphisms was disposed of in the discussion of $\mathbf{IPO}$. Lemma 5.3.2 easily ensures that the fibration construction preserves being countably-based. □

+§**5.3.7** When working with boundedly complete domains, which is the other case which we treat, we have a convenient way of moving from the fibre over a to that over b, namely *via* that over $a \curlywedge b$, and hence it is possible to form useful retracts.

Lemma Suppose $A \in \mathbf{bcCont}$ and let $f : Y \to A$, $g : Y \to X$ be arbitrary continuous functions. Then there is a continuous function $u : Y \to X$ such that $f = u\,;p$ and $\forall y.fy = p(gy) \Rightarrow uy = gy$.

Proof Let $y \in Y$. Put $x = gy$, $a = px$, $b = fy$ and $\gamma : a \curlywedge b \leq a$, $\delta : a \curlywedge b \leq b$. Define $uy = \Sigma\delta(\mathsf{P}\gamma x)$. We have to show that this is continuous, so let $y = \bigvee^{\uparrow} y_i$, *etc.*, so that the following diagram commutes:

$$\begin{tikzcd}
a & & b \\
& a \curlywedge b \arrow[ul, "\gamma"'] \arrow[ur, "\delta"] & \\
a_i \arrow[uu, "\alpha_i"] & & b_i \arrow[uu, "\beta_i"'] \\
& a_i \curlywedge b_i \arrow[ul, "\gamma_i"] \arrow[ur, "\delta_i"'] \arrow[uu, "\epsilon_i"] &
\end{tikzcd}$$

By functoriality, the corresponding diagram of fibres and substitutions ($\mathsf{P}a$ and $\mathsf{P}\gamma$, *etc.*) commutes, and indeed the three columns are (bi)limiting cones. By adjointness we also have $\Sigma\epsilon_i\,;\Sigma\delta = \Sigma\delta_i\,;\Sigma\beta_i$. Since A is continuous, $a \curlywedge b = \bigvee^{\uparrow}(a_i \curlywedge b_i)$. Finally recall that $x = \bigvee^{\uparrow} x_i$ in the display means that $x = \bigvee^{\uparrow} \Sigma\alpha_i x_i$ in its fibre. So we now have

$$\begin{aligned}
uy &= \Sigma\delta(\mathsf{P}\gamma x) && \text{definition} \\
&= \Sigma\delta\big(\textstyle\bigvee^{\uparrow} \Sigma\epsilon_i(\mathsf{P}\gamma_i x_i)\big) && \mathsf{P}(a \curlywedge b) = \mathsf{bilim}\,\mathsf{P}(a_i \curlywedge b_i) \\
&= \textstyle\bigvee^{\uparrow} \Sigma\delta[\Sigma\epsilon_i(\mathsf{P}\gamma_i x_i)] && \text{continuity} \\
&= \textstyle\bigvee^{\uparrow} \Sigma\delta_i(\mathsf{P}\gamma_i x_i) && \epsilon_i\,;\delta = \delta_i\,;\beta \\
&= \textstyle\bigvee^{\uparrow} \Sigma\beta_i(uy_i) && \text{definition} \\
&= \textstyle\bigvee^{\uparrow} uy_i && \mathsf{P}b \cong \mathsf{bilim}\,\mathsf{P}b_i
\end{aligned}$$

as required. The properties follow easily from the construction. $\square$

+§**5.3.8** We can now prove the result for boundedly complete domains.

Lemma $X \in \mathbf{bcCont}$ iff $A \in \mathbf{bcCont}$ and $\forall a \in A.\mathsf{P}a \in \mathbf{bcCont}$.

Proof

[$\Rightarrow$] $A \lhd X$, so clearly $A \in \mathbf{bcCont}$. Let $a \in A$; we use lemma 5.3.7 with $Y = X$, $f = \mathsf{K}a$ and $g = 1_X$, then the square

$$\begin{tikzcd}
X \arrow[r, "u"] \arrow[d] & X \arrow[d, "p"] \\
1 \arrow[r, "\ulcorner a \urcorner"] & A
\end{tikzcd}$$

commutes, so u factorises $u : X \to \mathsf{P}a \subset X$. Moreover the other property of u makes $\mathsf{P}a \lhd X$.

[$\Leftarrow$] We have only to check binary meets. Let $x \in \mathtt{P}a$, $y \in \mathtt{P}b$. Put $c = a \curlywedge b \in A$ and $\alpha : c \leq a$, $\beta : c \leq b$. Then $z = (\mathtt{P}\alpha x) \curlywedge (\mathtt{P}\alpha y) \in \mathtt{P}c$ is the meet. Note that it has to be in this fibre since p is a projection. For $w \in \mathtt{P}d$ with $w \leq x, y$, let $\alpha' : d \leq a$, $\beta' : d \leq b$, $\gamma : d \leq c$. Then $w \leq (\mathtt{P}\alpha' x) \curlywedge (\mathtt{P}\beta' y) = (\mathtt{P}\gamma(\mathtt{P}\alpha x)) \curlywedge (\mathtt{P}\gamma(\mathtt{P}\beta y)) = \mathtt{P}\gamma((\mathtt{P}\alpha x) \curlywedge (\mathtt{P}\beta y)) = \mathtt{P}\gamma z \leq z$ since $\mathtt{P}\gamma$ has a left adjoint. □

Proposition Fibred types form a class of display maps for **bcCont** and its flavours. □

$^+$§**5.3.9** Indeed this applies to other categories of (boundedly complete) domains.

Lemma Let $\mathcal{C}$ be a category of continuous boundedly complete domains and $\mathtt{P} : A \to \mathbf{IPO}^{cp}$ a continuous type-dependence with fibration $p : X \to A$. Then $X \in \mathcal{C}$ iff $A \in \mathcal{C}$ and $\forall a. \mathtt{P}a \in \mathcal{C}$.

Proof A and $\mathtt{P}a$ are retracts of X by lemma 5.3.8. We shall make $X \triangleleft A \times \prod_{a \in A} \mathtt{P}a$. Let $a \in A$; we shall make two uses of lemma 5.3.7. Let $u_a : X \to \mathtt{P}a$ as in the proof of lemma 5.3.8, and

$$\iota : X \hookrightarrow U \text{ by } x \mapsto (px, (u_a x : a \in A))$$

Put $Y = \mathtt{P}a \times A$, $f = \pi_1 : Y \to A$ and $g = \pi_0 : Y \to \mathtt{P}a \subset X$. Then we have $v_a : \mathtt{P}a \times A \to X$ continuous, where $v_a(x, b) = \Sigma\beta(\mathtt{P}\alpha x)$. Then by directed distributivity,

$$\pi : U \twoheadrightarrow X \text{ by } (a, (y_b : b \in A)) \mapsto \bigwedge(v_a(u_a x, px) : px \leq a)$$

is continuous. We have to show that $\iota \,;\, \pi = 1_X$. Let $x \in \mathtt{P}a \subset X$; then $\pi(\iota x) = \bigwedge(v_a(u_a, px) : px \leq a)$ is the inf of a set with least element x, since for $\alpha : px \leq a$, $v_a(u_a x) = \mathtt{P}\alpha(\Sigma\alpha x) \geq x$, and if $px = a$, $v_a(u_a x) = x$. □

Proposition Fibred types form a class of displays for any category of boundedly complete domains. □

$^+$§**5.3.10** The notion of a domain-indexed family of domains ought to generalise to spaces and locales, but I have not as yet been able to formulate the definitions in the absence of the specialisation order. I can, however, dispose of the use of Scott-continuity.

Lemma Let $A \in \mathbf{IPO}$ and $\mathtt{P} : A \to \mathbf{IPO}^{cp}$ be a functor. Then $\mathtt{P}$ is continuous iff

$$(\forall a \in A, x \in U \subset \mathtt{P}a : U \text{ open})\ (\exists a \in V \subset A : V \text{ open})\ (\forall \alpha : b \to a : b \in V)\ (\Sigma\alpha(\mathtt{P}\alpha x) \in U)$$

Proof This simply reformulates $\mathtt{P}a \cong \mathsf{bilim}(\mathtt{P}a' : a' \ll a)$ (specifically at the point x). U sets the degree of approximation to x which we require (by definition of Scott-openness, any directed sup which reaches x eventually gets into U, and conversely if this holds for all U then it reaches x). Then V is a sufficient stage in $\bigvee^{\!\!\uparrow}\{\Sigma\alpha(\mathtt{P}\alpha x) : \alpha : b \leq a, b \in V\}$ to get within U. □

5.4 Indexed Domain Theory

$^+$§**5.4.1** We have seen that continuous type-dependences, as defined in §5.1.12, behave very nicely *vis à vis* categories of domains, and that the alternative definitions fail to provide cartesian closure. In this section we demonstrate cartesian closure for continuous type-dependences; we have also to show that the exponentials remain in the given category of domains.

Theorem IPO is cartesian closed relative to fibred types.

Proof We have to construct the right adjoint, $\Pi\alpha$, to substitution, $\mathtt{P}\alpha$, over a display map $\alpha : B \to A$. By definition $B = \sum_A X$ for some $X \in \mathtt{P}A$, *i.e.* $X : A \to \mathbf{IPO}^{cp}$ is a continuous type-dependence with fibration $\alpha : B \to A$. Let $Y \in \mathtt{P}B$ and put $\beta : C = \sum_B Y \to B$ for the corresponding display map.

By our intuition from **Set**,

$$\Pi\alpha Y a \quad = \quad \{u : Xa \to Y : \forall b \in Xa. ub \in Yb\}$$

with the induced order from $[Xa \to Y]$; the least element is $\lambda b.\bot_{Yb}$ and $\bigvee^{\uparrow}$ is inherited by continuity. We have to show that $\Pi\alpha Y : A \to \mathbf{IPO}^{cp}$ is a continuous type-dependence and that $\mathtt{P}\alpha \dashv \Pi\alpha$.

First, for $\gamma : a' \leq a$, we have to define the substitution $\Pi\alpha Y\gamma$. Let $x' \in Xa'$ and $\delta : x' \leq \Sigma_X \gamma x'$, then

$$\Pi\alpha Y\gamma u x' \quad = \quad Y\delta[u(\Sigma_X\gamma x')]$$

which clearly lies in Yx' as required. To prove continuity in x' we have to use the continuity of $Y : B \to \mathbf{IPO}^{cp}$: if $x' = \bigvee^{\uparrow} x'_i$ with $\xi_i : x'_i \leq x'$ then the diagram

$$\begin{tikzcd}
Y(\Sigma_X\gamma x'_i) \arrow[r, "Y\delta_i"] & Yx'_i \\
u(\Sigma_X\gamma x) \in Y(\Sigma_X\gamma x') \arrow[u, "Y(\Sigma_X\gamma\xi_i)"] \arrow[r, "Y\delta"] & Yx' \arrow[u, "Y\xi"']
\end{tikzcd}$$

commutes, the vertical columns being (bi)limiting cones. Continuity in u is trivial.

$\Pi\alpha Y\gamma$ has a left adjoint given at $v \in \Pi\alpha Ya'$, $x \in Xa$ by

$$\Sigma_{\Pi\alpha Y}\gamma v x \quad = \quad \Sigma_Y\delta[v(X\gamma x)]$$

Recalling the contravariance of the function-space in its first argument, the following are equivalent:

$$\begin{array}{lcll}
\Sigma_{\Pi\alpha Y}\gamma v & \leq & u & \\
\Sigma_Y\delta[v(X\gamma x)] & \leq & ux & \forall x \in Xa \\
v(X\gamma x) & \leq & Y\delta(ux) & \forall x \in Xa \\
vx' & \leq & Y\delta[u(\Sigma_X\gamma x')] & \forall x' \in Xa' \\
v & \leq & \Pi\alpha Y\gamma u &
\end{array}$$

We now have a functor $\Pi\alpha Y : A \to \mathbf{IPO}^{cp}$, and we have to show that it is continuous. Let $a = \bigvee^{\uparrow} a_i$, $\gamma_i : a_i \leq a$, $x \in Xa$; then

$$\bigvee\nolimits^{\uparrow} \Sigma_{\Pi\alpha Y}\gamma_i(\Pi\alpha Y\gamma_i ux) \quad = \quad \bigvee\nolimits^{\uparrow} \Sigma_Y\delta_i[Y\delta_i[u[\Sigma_X\gamma_i[X\gamma_i x]]]] \quad = \quad ux$$

since X and Y are continuous.

Now we have to show that $\Pi\alpha$ is right adjoint to $\mathtt{P}\alpha$. Let $Z \in \mathtt{P}A$ with fibration $\gamma : D = \sum_A Z \to A$. As in §5.1.8, $f : \mathtt{P}\alpha Z \to_B Y$ correspond to $g : Z \to_A \Pi\alpha Y$ by $fbu = gaub$, where $b \in Xa \subset B$ and $u \in \mathtt{P}\alpha Zb = Za$. Bijectiveness and continuity follow immediately as in that section. □

+§**5.4.2** Since we have constructed a continuous type-dependence, $\Pi\alpha Y$, it is instructive to see the corresponding fibration. This is

$$\{\langle a, u\rangle : a \in A, u : Xa \to Y, \forall b \in Xa. \beta(ub) = b\}$$

with order relation

$$\langle a, u\rangle \leq \langle a', u'\rangle \quad \iff \quad a \leq a' \quad \wedge \quad (\forall b \in Xa, b' \in Xa' : b \leq b')(ub \leq u'b')$$

It would seem that we may make this construction even for projections, since we need only the display map and its left adjoint (and not the type-dependence) to define it. However this turns out not to work because of examples 5.2.8.

${}^{+}$§**5.4.3** Next we show that the indexed product construction is continuous w.r.t. bilimits in its various arguments.

Proposition Let $A = \mathsf{bilim}\, A_i$, $B = \mathsf{bilim}\, B_i$, $C = \mathsf{bilim}\, C_i$ and $\alpha_i : B_i \to A_i$, $\beta_i : C_i \to B_i$ be compatible families of fibred types. Let $\gamma_i : D_i \to A_i$ be the indexed product $\Pi\alpha_i\beta_i$. Let $\alpha : B \to A$, $\beta : C \to B$ and $\gamma : D \to A$ be the mediating maps, where $D = \mathsf{bilim}\, D_i$. Then $\gamma = \Pi\alpha\beta$, *i.e.* "$\Pi\alpha Y$" is continuous in A, α and Y.

Proof $\Pi\alpha\beta a$ is the ipo of u making the diagram

$$\begin{tikzcd} C \arrow[dr, "\beta"'] & Xa \arrow[l, "u"'] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & 1 \arrow[d, "\ulcorner a \urcorner"] \\ & B \arrow[r, "\alpha"] & A \end{tikzcd}$$

commute, where the square is a pullback. We have to show that such u's are given precisely by compatible families of similar (but subscripted) diagrams. We may see what the projection maps are by considering the commutative diagram

$$\begin{tikzcd} \Pi\alpha Y a \arrow[r, hook] \arrow[d] & {[Xa \to C]} \arrow[d] \\ \Pi\alpha_i Y_i a_i \arrow[r, hook] & {[X_i a_i \to C_i]} \end{tikzcd}$$

where the right column is (bi)limiting; we have to show that the left is too. We have only to observe that $u \in [Xa \to C]$ satisfies the condition iff its projects do. □

${}^{+}$§**5.4.4** We now restrict these results to the various categories of domains

Theorem $\mathbf{BiPos}_{\mathrm{f}}$ is cartesian closed relative to fibred types.

Proof By the results of §5.3, we have only to show that if A, Xa and Yb are in a given category of domains, $\mathcal{C} = \mathbf{BiPos}_{\mathrm{f}}$, (for $a \in A$, $b \in B = \sum_A X$) then so is $\Pi\alpha Y a$. This is trivial for finite domains, and proposition 5.4.3 allows us to introduce bilimits in all three arguments. Functoriality easily allows retracts to be introduced and the result to be extended to **ContDom**, but we were unable to show the indexed-fibred correspondence for this case. □

${}^{+}$§**5.4.5** For the case of **bcCont** and other categories of boundedly complete domains, we make $\Pi\alpha Y a \triangleleft [Xa \to C]$.

Theorem Any category of boundedly complete domains is cartesian closed relative to fibred types.

Proof Let $a \in A$, $B = \sum_A X$, $\beta : C \to B$ be as before. In lemma 5.3.7, let $Y = [Xa \to C] \times Xa$, $f = \pi_1 : Y \to Xa \subset B$ and $g = \mathsf{ev} : Y \to C$; then we have a continuous function $r : Y \to C$ with $p(r(u,b)) = b$ and if $p(u(b)) = b$ then $r(u,b) = u(b)$, so the exponential transpose of r makes $\Pi\alpha Y a$ a retract of $[Xa \to C]$. □

${}^{+}$§**5.4.6** By "Domain Theory" we mean more than exponentials, so we have to extend these things to the indexed case. However most of these things are very easy.

Proposition Let $\mathtt{P} : A \to \mathcal{C}^{cp}$ be a continuous type dependence with fibration $p : X \to A$.

(a) Let $f : X \to X$ be any fibred continuous function over A; then f has a least fibred fixpoint.

(b) There is a fibred embedding $X \rightarrowtail X_A^X$ over A.

(c) Any continuous functor $F : \mathcal{C}^{cp} \to \mathcal{C}^{cp}$ extends to a fibred endofunctor of the fibration of $\mathcal{C}$ over itself by means of these displays.

(d) We have fibred bilimits over A.

(e) There is a fibred embedding of X into a fibred η-model.

Proof

[a] This even holds for projections; we proved it in proposition 2.4.8.

[b] The proof of proposition 1.2.17b may be read indexedly.

[c] Its value at X is given by $\mathtt{P}\,;F$.

[d] Follows from proposition 5.2.6

[e] Apply (c) to (b,c) as in proposition 2.2.8b. □

5.5 Equality, Function-Spaces and Type-of-Types

§**5.5.1** In the context of category theory (and we course course take it as axiomatic that this is the appropriate foundational language for mathematics and theoretical computer science) the system of all "sets" is not just a set or type but a category. In the absence of a requirement for strong polymorphism it is generally accepted that this category should be an elementary topos as in §4.4.

Note The account given in this section is rather patchy, but was fully worked out in my joint paper with Andrew Pitts, *A Note on Russell's Paradox in Locally Cartesian Closed Categories*.

In our case we want not just a type, V, of types, but an *internal category*, with an object C_1 of morphisms as well as an object $C_0 = \mathsf{V}$ of objects. Thus we make the

Definition A *typos* is a "cartesian closed self-indexed", *i.e.* relatively cartesian closed category $(\mathcal{S}, \mathcal{D})$, with a "generic family", *i.e.* an internal category $\mathcal{C}$ in $\mathcal{S}$ together with an equivalence $\mathcal{S} \simeq \mathcal{C}$ of $\mathcal{S}$-indexed categories.

The original definition proposed by Andrew Pitts had "locally" for "relatively"; we devote the last part of this section to showing that such a gadget is necessarily degenerate.

§**5.5.2** Before working on this definition let us first unscramble it from the abstract way in which it is presented. We have therefore to recall what is an internal category (especially over a *relatively* cartesian closed category), how to make the two categories into fibrations and what it means for them to be equivalent.

The morphism-set of a category $\mathcal{C}$ is really a polymorphic notion: instead of the scheme $C_1 \rightrightarrows C_0$ we should really think in terms of the hom-set $\mathcal{C}(X, Y)$ as an object with two free variables X, Y of type C_0. From this we see that $\langle \mathsf{dom}, \mathsf{cod} \rangle : C_1 \to C_0 \times C_0$ must be a *display map*. This enables us to construct the pullback

$$\begin{tikzcd}
C_2 \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C_1 \times C_0 \arrow[d, "\langle \mathsf{dom}, \mathsf{cod} \rangle \times 1"'] \\
C_0 \times C_1 \arrow[r, "1 \times \langle \mathsf{dom}, \mathsf{cod} \rangle"'] & C_0 \times C_0 \times C_0
\end{tikzcd}$$

(as in §3.2.8) which yields the object of composable pairs, and hence the composition map $\mathsf{comp} : C_2 \to C_1$, and similarly formulate the associative law. Of course we also have $\mathsf{id} : C_0 \to C_1$ as usual.

Given an internal category we construct a fibration as in §4.2.10 whose objects over $A \in \mathcal{S}$ are $\mathcal{S}$-maps $X : A \to C_0$. The equivalence between $\mathcal{C}$ and $\mathcal{S}$ is to be (at least) a full and faithful assignment of a display $X \to A$ to such a map. Taking in particular the identity, $C_0 \to C_0$, we have a display $\mathsf{G} \to C_0$. The display corresponding to $X : A \to C_0$ is now given by the pullback $X^*\mathsf{G} \to A$ of $\mathsf{G} \to C_0$ against this map.

Proposition A typos is determined by a relatively cartesian closed category $(\mathcal{S}, \mathcal{D})$ together with a pair of displays $\mathsf{G} \to C_0$ and $\langle \mathsf{dom}, \mathsf{cod} \rangle : C_1 \to C_0 \times C_0$ and $\mathsf{f} : \mathsf{dom}^* \, \mathsf{G} \to \mathsf{cod}^* \, \mathsf{G}$ such that

(i) any display occurs (not uniquely) as a pullback of $\mathsf{G} \to C_0$ and

(ii) for any morphism $f : X^*\mathsf{G} \to Y^*\mathsf{G}$ (where $X, Y : A \rightrightarrows C_0$) there is a unique $\ulcorner f \urcorner : A \to C_1$ with $\ulcorner f \urcorner \,;\mathsf{dom} = X$, $\ulcorner f \urcorner \,;\mathsf{cod} = Y$ and

$$\begin{tikzcd}
X^*\mathsf{G} \arrow[r, "f"] \arrow[d, "\cong"'] & Y^*\mathsf{G} \arrow[d, "\cong"'] \\
\ulcorner f \urcorner^*(\mathsf{dom}^* \, \mathsf{G}) \arrow[r, "\ulcorner f \urcorner^* \mathsf{f}"'] & \ulcorner f \urcorner^*(\mathsf{cod}^* \, \mathsf{G})
\end{tikzcd}$$

commutes. □

$^+$§**5.5.3** This definition is highly redundant. Having (quite correctly) made the observation that we should take account of the morphisms into our notion of "type of types", we may forget it again. For given two "names" (elements of $C_0 = \mathsf{V}$) for sets, we already know how many morphisms there are to be between them, because we know how many functions there are between the sets they name. Alternatively, we observe that we are using the display $\mathsf{G} \to C_0$ to carve out a small *full* subcategory of $\mathcal{S}$.

Lemma

(a) $\langle \mathsf{dom}, \mathsf{cod} \rangle : C_1 \to C_0 \times C_0$ is the fibred exponential of $\mathsf{G} \to C_0$ (or rather $\mathsf{G} \times C_0 \to C_0 \times C_0$) by itself (*i.e.* $C_0 \times \mathsf{G} \to C_0 \times C_0$); more specifically, $C_1 = \sum_{u,v \in \mathsf{V}} \mathsf{G}[v]^{\mathsf{G}[u]}$.

(b) $\mathsf{dom}^* \, \mathsf{G}$ and $\mathsf{cod}^* \, \mathsf{G}$ are objects in the fibre over C_1, whose components at $f \in C_1$ are just $\mathsf{dom}\, f$ and $\mathsf{cod}\, f$; so $\mathsf{f} : \mathsf{dom}^* \, \mathsf{G} \to \mathsf{cod}^* \, \mathsf{G}$ is the morphism in this fibre whose component at f is just f. □

Corollary In proposition 5.5.2 we may delete (ii). □

Since C_1 will make no further explicit appearance, we shall revert to the name V for the object-of-objects (type-of-types) C_0.

V is not really so much a type of *types* but a type of *names* of types; then for each *name* $\ulcorner A \urcorner = v \in \mathsf{V}$ we have to prescribe the corresponding *type* $A = \mathsf{G}[v] = \mathbf{t}T^{-1}(\ulcorner A \urcorner)$, where $\mathbf{t}T : \mathsf{G} \to \mathsf{V}$. Commonly discussions of type of types omit this second datum; in the models we shall construct in the next section V itself will actually be a lattice and so carry very little information on its own. On the other hand, if we manage to present V as a *combinator* as in §5.5.5 and §5.6.9, then it does carry sufficient information on its own.

+§**5.5.4** We have still further redundancy. Since every display occurs as a pullback of $\mathsf{G} \to \mathsf{V}$ (and every pullback of a display is a display), we need not mention $\mathcal{D}$ explicitly. We need only require that all pullbacks of $\mathsf{G} \to \mathsf{V}$ exist and that this class of maps, once constructed, have the required properties. There does not, unfortunately, seem to be any abbreviation of axiom (ii) of §4.3.2 (which formally requires composability of displays but really concerns sums), but we can at least reduce the last two axioms:

Proposition A typos is determined by a category $\mathcal{S}$ and a map $\mathsf{G} \to \mathsf{V}$ such that

(i) The pullback of $\mathsf{G} \to \mathsf{V}$ against any map exists in $\mathcal{S}$ (write $\mathcal{D}$ for the class of such pullbacks)

(ii) $\mathcal{D}$ is closed under composition,

(iii) Any terminal projection is in $\mathcal{D}$,

(iv) V has a binary operation $\to: \mathsf{V} \times \mathsf{V} \to \mathsf{V}$ whose pullback against $\mathsf{G} \to \mathsf{V}$ is the fibred exponential of $\mathsf{G} \to \mathsf{V}$ with itself. □

The pullback in the last case is of course just C_1 again. We shall look again at the definition in section 6, where we shall replace (iii) by the following:

Lemma Let $\mathcal{S}$ have a typos structure $\mathsf{G} \to \mathsf{V}$. Then V has an element $\ulcorner 1 \urcorner$ and a binary operation $\times$ giving rise to the finite products in $\mathcal{S}$. □

§**5.5.5** To define a typos we therefore need only give a single morphism in a category. Here is the best known example.

Theorem The closures in $P\omega$ provide a typos structure for $\mathbf{AlgLat}_\omega$.

Proof Put $\Lambda = P\omega$, $\mathsf{V} = \|\lambda a. \bigvee_n (1 \curlyvee a)^n\|$ and $\mathsf{G} = \|\lambda u. \langle \mathsf{V}u_0, \mathsf{V}u_0 u_1 \rangle\|$; $\mathsf{G} \to \mathsf{V}$ is given by the left component.

The class $\mathcal{D}$ of pullbacks of $\mathsf{G} \to \mathsf{V}$ contains the terminal projection $A = \|a\| \to 1$ by considering $\ulcorner a \urcorner : 1 \to \mathsf{V}$. We construct pullbacks as in §5.1.6, since $\mathsf{G} = \sum_\mathsf{V} \mathsf{V}$. For the exponential we only need $\mathsf{G}^\mathsf{G}_{\mathsf{V}\times\mathsf{V}} = \|\lambda u. \langle \langle \mathsf{V}u_{00}, \mathsf{V}u_{01} \rangle, \lambda x. \mathsf{V}u_{01}(u_1(\mathsf{V}u_{00}x)) \rangle\|$.

For composites let $f : A \to \mathsf{V}$, $g : B \to \mathsf{V}$ with pullbacks $B = \sum_A f \to \mathsf{G}$ and $C = \sum_B g \to \mathsf{G}$. Put $h = \lambda au. \langle fau_0, g(fau_0)u_1 \rangle$. Then the composite $C \to B \to A$ is the pullback of $\mathsf{G} \to \mathsf{V}$ along h. □

The combinatorial flavour of this suggests that one might try to formulate a new (λ-indefinable but relatively consistent) combinator V. For instance $\mathsf{V} = \mathsf{VV} = \mathsf{QVV}$ says that V is a V-type of V-types; to this we add axioms forcing cartesian closure.

In the next section a new typos will be constructed from fibred types.

+§**5.5.6** Both this traditional example and the new ones to be constructed fail to be *extensional* in the sense that there are continuous functions $\mathsf{V} \to \mathsf{V}$ which do not preserve isomorphisms.

We have already foresaken the traditional belief in Category Theory that all mathematical constructions are functors. However even in the known cases of nonfunctorial constructions we at least have preservation of isomorphisms, so is there some serious problem here?

Perhaps not. Isomorphism, recall, is a *structure*, not a *property*, and we have no reason to suppose that even when an isomorphism between to objects exists then it must be recursively realisable (in whatever sense). It is therefore plausible that a function, and even one with type values, may distinguish between accidentally isomorphic objects.

On the other hand, this was the reason for throwing isomorphisms into definition 5.1.12. Any continuous functor $\mathcal{C}^{cp} \to \mathcal{C}^{cp}$ necessarily preserves isomorphisms, and so if we had a model with $\mathsf{V} = \mathcal{C}^{cp}$ we would have extensionality. This cannot be done classically, but

Question Does the Effective Topos provide an Extensional Typos?

$^{+}$§**5.5.7** The notion of typos clearly allows interpretation of *function spaces* and *type of types*. We devote the remainder of this section to the demonstration that if we add *equality* to these two then we have inconsistency. However it is very interesting to observe that there is apparently no *pairwise* inconsistency amongst these concepts.

Clearly "classical" categorical model theory (the interpretation of the logic of equality, predicate calculus, quantifiers and so on in categories, in particular toposes) allows nondegenerate interpretations of equality and function spaces.

Our categories of domains allow interpretation of function spaces and type of types.

Joyal's Arithmetic Universes appear in some sense to allow interpretation of equality and type of types. (Unfortunately no account of this important work has appeared on paper.)

$^{+}$§**5.5.8** A typos with equality would be just too good: it falls over under its own weight. Let $\mathcal{S}$ be a locally cartesian closed category which is equivalent as an $\mathcal{S}$-indexed category to an internal category $\mathcal{C} = (C_0, C_1)$. For X over A in $\mathcal{S}$ we may construct the object

$$\mathsf{Mono}_A(X) \quad = \quad \left\{ U \xrightarrow{\ i\ } X \ :\ \left(\forall V \underset{f}{\overset{g}{\rightrightarrows}} U : f\,;i = g\,;i\right)(f = g)\right\}$$

as a subobject of $C_0 \times C_1$ by means of (pullbacks,) equalisers and a Π functor.

Then $i : U \hookrightarrow X$ over $\mathsf{Mono}_A(X) \to A$ is a generic (saturated) mono in $\mathcal{S}$, so that $\mathcal{S}$ is (locally cartesian closed and) weakly well powered.

If Question 4.4.12 has a positive answer then $\mathcal{S}$ is a topos, and has a subobject classifier Ω.

On the other hand, we have a "big" set G. Any object A of course gives rise to a display $A \to 1$ which is a pullback of $\mathsf{G} \to \mathsf{V}$ against some $1 \to \mathsf{V}$; the remaining map in the square is a mono $A \hookrightarrow \mathsf{G}$.

But putting $A = \Omega^{\mathsf{G}}$ we deduce by Cantor's theorem (corollary 1.5.13) that $\mathcal{S}$ is degenerate.

Note This contradiction is examined in detail in *A Note on Russell's Paradox in Locally Cartesian Closed Categories.*

§**5.5.9** It is possible to dispose of the use of Ω and bring back the diagonal argument underlying Cantor's theorem directly into the proof itself with V or G playing the rôle previously played by Ω. It would be nice if we could remove equality from this proof and show positively that any object of any typos is inhabited. This would almost certainly show that any typos gives rise to a model of the λ-calculus.

Theorem Let $\mathcal{S}$ be a locally cartesian closed category with a generic family. Then $\mathcal{S}$ is degenerate.

Proof (Pitts) First, there is a pullback square

$$\begin{tikzcd}
\mathsf{V}^{\mathsf{G}} \arrow[r, "i"] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & \mathsf{G} \arrow[d] \\
1 \arrow[r, "\ulcorner \mathsf{V}^{\mathsf{G}} \urcorner"] & \mathsf{V}
\end{tikzcd}$$

We perform the rest of the argument in an indexed rather than fibred style, in which $\mathsf{G}[v]$, for instance, denotes the inverse image of v in G. For $A \in \mathcal{S}$, the equality type indexed by $A \times A$ corresponding to the diagonal $A \to A \times A$ will be written $\Delta_A[a_1, a_2]$.

Let X be a fixed object of $\mathcal{S}$ and consider

$$\left(\sum_{f \in \mathsf{V}^{\mathsf{G}}} \Delta_{\mathsf{G}}[g, if] \ \times\ X^{\mathsf{G}[f(if)]} \ :\ g \in \mathsf{G} \right)$$

which is a G-indexed family of types, so we write its display as a pullback of $\mathsf{G} \to \mathsf{V}$ along some $\xi : \mathsf{G} \to \mathsf{V}$. Thus the expression above is isomorphic to $\mathsf{G}[\xi g]$.

Now ξ gives a global element of V^{G}, to which one may apply $i : \mathsf{V}^{\mathsf{G}} \hookrightarrow \mathsf{G}$ to obtain a global element $i\xi$ of G. Substituting this for g in the isomorphism yields

$$\begin{aligned} \mathsf{G}[\xi(i\xi)] &\cong \textstyle\sum_f \Delta_{\mathsf{G}}[i\xi, if] \times X^{\mathsf{G}[f(if)]} \\ &\cong \textstyle\sum_f \Delta_{\mathsf{V}^{\mathsf{G}}}[\xi, f] \times X^{\mathsf{G}[f(if)]} \\ &\cong X^{\mathsf{G}[\xi(i\xi)]} \end{aligned}$$

Thus writing Y for the object $\mathsf{G}[\xi(i\xi)]$ we have $Y \cong X^Y$ so X is reflexive in $\mathcal{S}$ and by proposition 1.5.6 is inhabited.

We have shown that every object of a certain cartesian closed category with all finite limits in inhabited. Hence by proposition 1.5.11 $\mathcal{S}$ is degenerate. □

§**5.5.10** Here for the record are Andrew Pitts' comments on this proof.

"Although it may not appear so at first, the above argument is (an elementary version of) the one given [in §5.5.8]: the necessity of showing $\mathcal{S}$ has a subobject classifier Ω is circumvented by systematically replacing Ω by V in [that] argument, at the same time replacing negation $\neg(-)$ by the exponential $X^{(-)}$, for X any fixed object. Thus the above proof is really Russell's Paradox in disguise! To make this clearer, let me remind you of the following argument [corollary 1.5.13]:

If a topos $\mathcal{E}$ contains an object T for which there is a monomorphism $i : PT = \Omega^T \hookrightarrow T$, then $\mathcal{E}$ is degenerate. For consider the subobject of T given by

$$(\exists s \in PT)(t = is \wedge \neg(is \in s))$$

(*cf.* the object considered in §5.5.9) and let $\xi : 1 \to PT$ be the corresponding global element. Then one has in $\mathcal{E}$:

$$\begin{aligned} i\xi \in \xi &\iff (\exists s \in PT : i\xi = is)\neg(is \in s) \\ &\iff (\exists s \in PT : \xi = s)\neg(is \in s) \\ &\iff \neg(is \in s) \end{aligned}$$

which leads to a contradiction.

Note that one could dually start with the subobject of T given by

$$(\forall s \in PT)\neg(t = is \wedge \neg is \in s)$$

and derive a similar contradiction. Taking the analogue of this in type theory (with $X^{(-)}$ for $\neg(-)$ and V for Ω, *etc.*) probably gives another proof of the result with the technical advantage that the proof is *completely within the* $(=,\Pi)$ *fragment of Martin-Löf type theory.* $\mathsf{G} = \sum_{v\in\mathsf{V}} \mathsf{G}[v]$ is not in this fragment, but only occurs *negatively* in the argument: $\mathsf{V}^{\mathsf{G}} \cong \prod_{v\in\mathsf{V}} \mathsf{V}^{\mathsf{G}[v]}$, *etc.*"

Indeed, writing $\nabla^X_A[a,b] = X^{\Delta_A[a,b]}$, we construct the object

$$\prod_{f\in\mathsf{V}^{\mathsf{G}}} \nabla^X_{\mathsf{G}}[g, if]^{X^{\mathsf{G}[f(if)]}} \quad\cong\quad X^{\mathsf{G}[\xi g]} \quad=\quad \mathsf{G}[\theta g]$$

then with $Z = \mathsf{G}[\theta(i\theta)]$, $Z \cong X^Y \cong X^Z$ as before. Since in some sense

$$\nabla^X_A[a,b] \quad=\quad \begin{cases} X \text{if } a = b \\ 1 \text{ otherwise} \end{cases}$$

it has more chance of existing in a category in which every object is inhabited, so perhaps this leads to the "positive" result suggested at the beginning of §5.5.9.

5.6 Type of Types in Domains

+§**5.6.1** Now we are in a position to construct a type of types in *any* category of domains $\mathcal{C}$, subject to the difficulties we have encountered in the past in manipulating these. Clearly we shall only be able to do this for *small* categories.

From proposition 5.5.4 we have only to find a fibred type $\mathsf{G} \to \mathsf{V}$ in $\mathcal{C}$ of which any other is a pullback. Since pullback of fibration corresponds to precomposition of indexation (lemma 4.2.13) it suffices to find the following:

Definition Let $\mathcal{C}$ be a small category of domains and $\mathbf{t}T : \mathsf{V} \to \mathcal{C}^{cp}$ a type dependence (§5.1.12). This is *saturated* if given any other type dependence $X : A \to \mathcal{C}^{cp}$ there is a continuous map $\ulcorner X \urcorner : A \to \mathsf{V}$ and a natural isomorphism $u : X \cong \ulcorner X \urcorner \,;\mathbf{t}T$.

Lemma Let $\mathbf{t}T : \mathsf{V} \to \mathcal{C}^{cp}$ be a saturated type dependence and $\mathsf{G} \to \mathsf{V}$ the corresponding fibration. Then $\mathsf{G} \to \mathsf{V}$ is a generic family, making $\mathcal{C}$ a typos.

Proof Let $p : X \to A$ be a display map, *i.e.* a fibred type corresponding to a continuous type-dependence $\mathsf{P} : A \to \mathcal{C}^{cp}$. Then by hypothesis we have $\ulcorner X \urcorner : A \to \mathsf{V}$ and $u : \mathsf{P} \cong \ulcorner X \urcorner ; \mathbf{t}T$. The latter is an indexed functor, so corresponds to a fibred equivalence between the corresponding displays, $u : X \cong \ulcorner X \urcorner^* \mathsf{G}$. Hence $p : X \to A$ occurs as a pullback of $\mathsf{G} \to \mathsf{V}$ as required. □

+§**5.6.2** This brings us directly back to the similarity of $\mathcal{C}^{cp}$ with its objects, upon which we first remarked in §2.2.7 (though there we mentioned $\mathbf{IPO}^{em}$). $\mathcal{C}^{cp}$ is a category rather than a poset, but it is possible (though beyond what could be done within the scope of this work) to generalise domain theory in this way. Having done this we would like to put $\mathsf{V} = \mathcal{C}^{cp}$.

The saturated type-dependence is then of course the identity $\mathcal{C}^{cp} \to \mathcal{C}^{cp}$. *This is clearly the correct approach, and it fails in our case* only *because of the inadequacy of classical logic.* We may seriously expect to be able to proceed in this fashion in the Effective Topos.

A more significant problem than its algebraic structure is that $\mathcal{C}^{cp}$ only has *countable* filtered colimits. Since it comes quite close to having binary coproducts, Freyd's paradox (proposition 1.5.8) means that we cannot get round this problem, at least in a classical world.

We shall, however, be able to "cover" $\mathcal{C}^{cp}$ with a suitable domain V to satisfy the above definition. We do *not* require the map $A \to \mathsf{V}$ to be unique; to do so would on the one hand force $\mathsf{V} \simeq \mathcal{C}^{cp}$ and on the other amount to *equality* on objects of $\mathcal{C}$. It is this covering operation which loses the extensionality mentioned in §5.5.6.

+§**5.6.3** Of course as already remarked we have to have a *small* category of domains. However we need a better grasp of the object-set than the usual "up to equivalence", so after, or rather in the spirit of, theorem 2.6.12, we may as well assume $\mathcal{C} = \mathbf{Retr}(\Lambda)$ with Λ algebraic. By "concrete domain" we shall here mean the image of a coclosure of Λ, so that there are canonical embeddings and projections between them. We have to do somewhat more work to cut down our indexed domains to a "canonical" form.

In §2.6.11 we approximated domains by means of bilimits of *embeddings*, and we shall do the same here. However does the fact that we are interested in diagrams of *comparisons* matter? (In a filtered diagram of monos we have no parallel pairs of maps.) Of course not, because we have just confused the objects (diagrams of comparisons) with their approximating morphisms ("natural embeddings").

Bilimits raise a particular question of canonicity, since we wish to have directed sups defined up to equality and not isomorphism (and anyway the whole point of the construction on which we are embarking is to discard a lot of isomorphisms). We have to choose a particular construction for (bi)limits, and we shall say that the *set-theoretic limit* of a diagram (X_i) is the set of compatible families (x_i). This approach is facilitated where we can express things as bilimits or directed sups of "atomic" (*i.e.* finite) things, and this is precisely what algebraicity is about.

Definition A *canonically indexed domain* is a continuous type-dependence $\mathtt{P} : A \to \mathcal{C}^{cp}$ where A is algebraic and

$$\mathtt{P}a \text{ is } \begin{cases} \text{a concrete domain} & \text{if } a \in A_{fp} \\ \mathsf{bilim}\{\mathtt{P}b : b \in A_{fp} \cap \downarrow a\} & \text{otherwise} \end{cases}$$

where the limit is the "set-theoretic" one. This enables us to restrict attention to A_{fp}.

The restriction to algebraic A is a convenience to get round the Axiom of Foundation in set theory and other technical points: we could generalise to arbitrary ipos (at least continuous ones, at any rate) by introducing bases, but this definition serves our present purposes adequately. We shall regard $\mathtt{P}a$ as part of the data for canonical indexed domains only for $a \in A_{fp}$; the non-compact points are redundant and cause unnecessary complications later on.

$^{+}$§**5.6.4** Since continuous type-dependences are functors, of course morphisms between them are natural transformations. The canonicity programme extends equally to these: instead of arbitrary comparisons between domains we are interested in canonical embeddings, and so we define an *order relation* between canonical indexed domains, $\mathtt{P} \le \mathtt{Q}$. This holds if $\mathtt{P}a \le \mathtt{Q}a$ as coclosures on Λ for each $a \in A_{fp}$ (the bilimits take care of themselves) and the square of projections and homomorphisms

$$\begin{tikzcd}
\mathtt{Q}b \arrow[d] & \mathtt{Q}a \arrow[l, "\mathtt{Q}\alpha"'] \arrow[d] \\
\mathtt{P}b & \mathtt{P}a \arrow[l, "\mathtt{P}\alpha"'] \\
b \arrow[r, "\alpha"] & a
\end{tikzcd}$$

(or alternatively the corresponding square of embeddings and comparisons) commutes.

Lemma $L(A)$, the set of canonical A-indexed domains, is an ipo.

Proof First observe that there can indeed be at most one instance of the order relation between any two canonical domains, so we have a poset and not a category. The ipo structure is *created* by the forgetful functor $L(A) \to \mathsf{Cocl}(\Lambda)^{|A|}$, which means that to calculate it we need only consider the domains and not the comparisons.

Let $\mathtt{P}_i \in L(A)$ be a directed set; recall that the order relation is expressed by a commutative diagram as in the previous paragraph. We take the bilimits of the columns of this diagram; these correspond to directed sups in $\mathsf{Cocl}(\Lambda)$. The mediating maps give the comparisons in the bilimiting indexed domain.

If we had not restricted attention to A_{fp}, this proof would have become more complicated, and if we had not introduced canonicity, the directed sup would not be unique.

The bottom element of $L(A)$ has all its objects $\perp_{\mathsf{Cocl}(\Lambda)}$ and morphisms the unique comparison $1 \to 1$. $\square$

For technical reasons (arising from the problem with theorem 2.2.14), we shall find it convenient also to consider continuous indexations; the foregoing discussion applies equally to them (with the exception of the alternative form of the above diagram). Write $M(A)$ for the poset of canonically continuous A-indexations and canonical embeddings; $M(A)$ is also an ipo.

$^{+}$§**5.6.5** The main thing to show is that $L(A) \in \mathbf{bcCont}_\omega$. We have to use $M(A)$ to show that it is countably based.

Lemma Let $\mathtt{P} \in L(A)$. Then the forgetful functor $L(A) \to \mathsf{Cocl}(\Lambda)^{|A_{fp}|}$ (given by the coclosures naming the objects of an indexed domain), restricted to $\downarrow\mathtt{P} \subset L(A)$, has postinverse adjoints on both sides. With the corresponding problem for $M(A)$, there is a postinverse right adjoint.

Proof We have to show that given an arbitrary $\{W_a : a \in |A_{fp}|\}$ there is (for the left) a least $\mathtt{Q} \leq \mathtt{P}$ in $L(A)$ with $W \leq \mathtt{Q}$ and (for the right) a greatest $\mathtt{Q} \geq \mathtt{P}$ in $L(A)$ or $M(A)$ with $W \geq \mathtt{Q}$.

In the right case, for $\alpha\,; b \leq a$, we want the diagram in §5.6.4 to commute. Hence $\mathtt{Q}b \subset \mathtt{P}\alpha(\mathtt{Q}a)$. But $\mathtt{Q}b \subset W_b$, so

$$FWb \quad = \quad \inf_{\mathsf{Cocl}(\Lambda)}\{\mathtt{P}\alpha(W_a) : \alpha : b \leq a\} \quad \supset \quad \mathtt{Q}b$$

We may put equality to give the required postinverse right adjoint.

For the left case we consider the adjoint diagram to that in §5.6.4, and get $FWa = \bigvee_{\mathsf{Cocl}(\Lambda)}\{\Sigma\alpha(W_b) : \alpha : b \leq a\}$. □

Corollary

(a) $\downarrow\mathtt{P}$ is a countably based continuous lattice.

(b) $L(A)$ and $M(A)$ are in **bcCont**.

Proof

[a] They are retracts of $\mathsf{Cocl}(\Lambda)^{|A_{fp}|}$.

[b] An ipo is continuous iff every $\downarrow$-set is, and is boundedly complete iff every $\downarrow$-set is a lattice. Unfortunately we cannot deduce anything about size. □

$^{+}$§**5.6.6** We shall now construct the saturated type-dependence in two particular cases, namely $\mathbf{ContLat}_\omega$ and $\mathbf{bcCont}_\omega$. The methods can probably be extended to other categories, but these suffice to make the point.

Lemma

(a) There is a continuous indexation Ψ into which any canonically indexed domain may be embedded.

(b) $L(A) \in \mathbf{bcCont}_\omega$.

Proof

[a] Consider the functor $M(A) \to \mathcal{C}^{op}$ as a diagram of projections, and take its limit by theorem 2.2.14. We cannot replace "continuous indexation" by "continuous type-dependence" because as remarked there we do not have limits in $\mathcal{C}^{cp}$ as a functor.

[b] $L(A)$ is equivalent to a Scott-closed subset of $\downarrow\Psi$, where Ψ is a *canonically* continuous indexation isomorphic to the one just constructed. □

Proposition (a) $\mathbf{ContLat}_\omega$ and (b) $\mathbf{bcCont}_\omega$ have saturated type-dependences.

Proof

[a] With $A = \Lambda$, the limit in the lemma *may* be constructed in $\mathbf{ContLat}_\omega^{hm}$, so put $L = \downarrow\Psi$.

[b] We have $L = L(\Lambda) \in \mathbf{bcCont}_\omega$.

In either case we have a functor $L \to (\mathcal{C}^{cp})^\Lambda$ and hence $\mathbf{t}T : \mathsf{V} = L \times \Lambda \to \mathcal{C}^{cp}$. Let $\mathtt{P} : A \to \mathcal{C}^{cp}$ be a continuous type-dependence and suppose $(i, p) : A \triangleleft \Lambda$. Let $\mathtt{Q} \in L$ be isomorphic to $p\,; \mathtt{P} : \Lambda \to A \to \mathcal{C}^{cp}$, say $u : F(\Lambda)(\mathtt{Q}) \cong (p\,; \mathtt{P})$. Then with $\ulcorner\mathtt{P}\urcorner = \langle \mathsf{K}\mathtt{Q}, i\rangle : A \to L \times \Lambda = \mathsf{V}$ we have $u : \ulcorner\mathtt{P}\urcorner\,; \mathbf{t}T \cong \mathtt{P} : A \to \mathcal{C}^{cp}$ in the manner of definition 5.6.1. □

In this case $\bot \in \mathsf{V}$ is mapped to the trivial indexation, although in general it is consistent with the definition that its image be any domain with $\top$. Nevertheless it is convenient to assume that it does map to 1.

$^+$§**5.6.7** It is appropriate to remind ourselves of what we have proved.

Theorem There are nontrivial small categories which are relatively cartesian closed and have an internal category to which they are internally equivalent. In other words, the definition of a *typos* given in §5.5 is consistent. □

$^+$§**5.6.8** Having achieved this, the main aim of the work, we are in a position to pick up certain other questions.

First, we may replace Λ by some other model with the same category of retracts but for which the displays constructed in §5.1.6 are precisely the fibred types of the category. As usual we manufacture the new model from the biggest object in sight, namely the generic type G. It would seem reasonable to assume that this kind of construction ought to be possible in any (abstract) typos, but I have no idea how to find the retractions needed in general.

Secondly, we pick up the internal combinatory algebra structure on a λ-model from theorem 1.3.13 and construct the internal version of the category of retracts on it. Though $\mathsf{I} = \Lambda$ carries this structure and I^{I} an internal monoid structure, it is not possible to perform the Karoubi construction in $\mathbf{Retr}(\Lambda)$ unless we already have a type-of-types. This is because we have no way of constructing the object of idempotent elements of Λ.

It is, of course, possible to perform this construction without the type-of-types in a locally cartesian closed category (or topos) $\mathcal{E}$ in which $\mathbf{Retr}(\Lambda)$ is fully embedded preserving products and exponentials (*e.g.* by the Yoneda embedding). We return to such embeddings in §5.8.

$^+$§**5.6.9** This ties §5.1 and §5.2 together.

Theorem Let $\mathcal{C}$ be a small category of boundedly complete domains. Then there is a saturated object Λ such that every continuous type-dependence factors up to isomorphism through $\mathsf{Idem}(\Lambda) \to \mathcal{C}^{cp}$ (*cf.* 5.1.12) and hence can be represented as an element $X \in \Lambda$ with $X = \mathsf{P}AX = \mathsf{Q}XX$.

Proof Let $t : \mathsf{G} \to \mathsf{V}$ be a generic family for $\mathcal{C}$; as the biggest object in sight, we put $\Lambda = \mathsf{G}$. We already have $\mathsf{V} \triangleleft \mathsf{G}$, since any fibred type is in particular a retract, but so far V is only defined up to isomorphism (indeed less than that) and it will be convenient to replace it with an isomorphic copy later. Any $U \in \mathcal{C}$ may be expressed as a retract of G since it occurs as a fibre (namely that over $\ulcorner U \urcorner$) and by lemma 5.3.8 the fibres are retracts of the display in the boundedly complete case. In particular $\Lambda^\Lambda \triangleleft \Lambda$ and so we may put a λ-model structure on Λ.

Now we use lemma 5.3.7 with $Y = \mathsf{V} \times \Lambda$, $f = \pi_0$, $g = \pi_1$, so there is a $u : \mathsf{V} \times \Lambda \to \Lambda$ such that $u\,;t = \pi_0$ and $(\forall a \in \mathsf{V})(\forall x \in \mathbf{t}Ta)(u(a,x) = x)$. Hence we have $U : V \to \Lambda^\Lambda \subset \Lambda$ such that Ua is an idempotent for each a and has underlying set $\mathbf{t}Ta$. We also have, for $\alpha : b \leq a$, $Ub(Uax) = \mathbf{t}T\alpha(Uax)$ and $Ua(Ubx) = \Sigma_T\alpha(Ubx)$ by construction. Hence $\|U-\| = \mathbf{t}T : \mathsf{V} \to \mathcal{C}^{cp}$.

Now $U : \mathsf{V} \hookrightarrow \Lambda$ has a postinverse, $R = \lambda a.t(a\bot)$, since $R(Ua) = t(Ua\bot) = a$. Since, as remarked, we have used V at most up to isomorphism, and now we have an explicit and convenient expression for it as a retract of Λ, we may replace it by its image under this. Hence $\mathsf{V} = \mathsf{P}RU \in \mathsf{Idem}(\Lambda)$ and $\|-\| = R\,; \mathbf{t}T : \|\mathsf{V}\| \to \mathcal{C}^{cp}$. Then we have $\mathsf{V} = \mathsf{PVV} = \mathsf{QVV}$, and also, since $\mathsf{V} \to \mathcal{C}^{cp}$ was a saturated type-dependence, any type-dependence factors through $\|\mathsf{V}\| \subset \mathsf{Idem}(\Lambda)$ as required. □

Unfortunately I cannot see how to make $\mathsf{V} \in \|\mathsf{V}\|$, *i.e.* $\mathsf{V} = \mathsf{VV}$.

$^+$§**5.6.10** Finally we repeat the Karoubi construction.

Proposition We may express the notions of "category", "category with products" and "cartesian closed category" in any category with a class of display maps. *The base category does not itself need to be cartesian closed.*

Proof This is what §3.2 and §4.2 were all about! Being cartesian closed is an equational condition on (the hom-objects of) an internal category; at no point do we consider the collection of all maps between two objects in the base category. $\square$

Theorem Let Λ be a β-model in $\mathbf{bcCont}_\omega$ and $\mathcal{C} = \mathbf{Retr}(\Lambda)$. Then there is an internal category $\mathcal{C}'$ in $\mathcal{C}$ which is equivalent to $\mathcal{C}$ as a $\mathcal{C}$-category and universally splits idempotents in the internal monoid I^{I}.

Proof $\mathcal{C}$ is a typos by theorem 5.6.7. Let $\mathcal{C}'$ be the internal category of §5.5.2; this is equivalent to $\mathcal{C}$ as a $\mathcal{C}$-category by definition of a typos. I^{I} is an internal monoid of $\mathcal{C}$ and hence embedded in $\mathcal{C}'$ by the equivalence. We split idempotents internally in $\mathcal{C}'$ or $\mathcal{C}$ just as we do externally. Let $\mathsf{I}^{\mathsf{I}} \to \mathcal{D}$ be a functor between $\mathcal{C}$-categories in which $\mathcal{D}$ splits idempotents, then construct the mediating functor $\mathcal{C} \simeq \mathcal{K}(\mathsf{I}^{\mathsf{I}}) \to \mathcal{D}$ externally and verify that it yields a cartesian functor over $\mathcal{C}$ and hence (the unique, up to isomorphism) internal functor $\mathcal{C}' \to \mathcal{D}$ making the triangle commute. $\square$

5.7 Interpretation of Polymorphism

§**5.7.1** We have now completed the major constructions of the work, and we shall finish off with a few observations (rather than serious applications) about polymorphism. We have, in fact, provided two slightly different ways of approaching this, namely direct use of the indexed domain theory introduced in the first half of the present chapter, and *via* the "type-of-types" co just constructed. We shall find that the latter falls rather short of our expectations, and that whilst we have satisfied the definition posed by Andrew Pitts (§5.5.1), this particular programme for constructing a type of types has failed.

It is, however, worthwhile looking at the way in which the two forms of recursive construction introduced in §2.2, namely for fixed *points* and recursive *domains* are unified by the presense of a type of types, and also at certain other at first sight anomalous domain theoretic constructions which are explained in the context of indexed domain theory.

As we have already remarked in §3.5.5, in the presence of a type-of-types we no longer have a distinction between types and values. I would consider the two-sortedness of languages such as those introduced in §3.4.2 to be an aesthetic objection to them. This means that the Tarski fixpoint constructions in domains discussed in §1.4 and §2.1 are unified with the Plotkin-Scott-Smyth techniques for solving recursive domain equations in categories (§2.2). That there should be such a unification can hardly be a surprise, given the similarity of the methods.

+§**5.7.2** Let us begin by looking at domain constructions (endo*functors* of the "modified" category of domains $\mathcal{C}^{cp}$) as endo*functions* of the type of type, V.

Proposition Let $\mathcal{C}$ be a small category of domains and $F : \mathcal{C}^n \times (\mathcal{C}^{op})^m \to \mathcal{C}$ any continuous functor. Then there is a continuous function $\phi : \mathsf{V}^{n+m} \to \mathsf{V}$ making the following diagram commute up to isomorphism:

$$\begin{tikzcd}
\mathsf{V}^{n+m} \arrow[r] \arrow[d, "\phi"'] & (\mathcal{C}^{cp})^{n+m} \arrow[r] \arrow[d] & \mathcal{C}^n \times (\mathcal{C}^{op})^m \arrow[d, "F"] \\
\mathsf{V} \arrow[r] & \mathcal{C}^{cp} \arrow[r] & \mathcal{C}
\end{tikzcd}$$

Proof The existence of the middle map follows essentially from proposition 2.2.7; that of ϕ is now just the definition of a saturated type-dependence (§5.6.1). $\square$

Question Is there a converse to proposition 2.2.7, *i.e.* given a continuous functor $\Phi : \mathcal{C}^{cp} \to \mathcal{C}^{cp}$ is there a functor $F : \mathcal{C} \times \mathcal{C}^{op} \to \mathcal{C}$ making the following diagram commute?

$$\begin{tikzcd}
\mathcal{C}^{cp} \arrow[r, "\Delta"] \arrow[d, "\Phi"'] & \mathcal{C}^{cp} \times \mathcal{C}^{cp} \arrow[r] & \mathcal{C} \times \mathcal{C}^{op} \arrow[d, "F"'] \\
\mathcal{C}^{cp} \arrow[rr] & & \mathcal{C}
\end{tikzcd}$$

$^{+}$§**5.7.3** What about recursive domain equations?

Proposition Let F be a continuous co- or contravariant endofunctor of a small category of domains and $\phi : \mathsf{V} \to \mathsf{V}$ the corresponding endofunction constructed in proposition 5.7.2.

(a) Let $\ulcorner A \urcorner$ be a fixpoint of ϕ with corresponding type $A \in \mathcal{C}$. Then $FA \cong A$.

(b) Let $\eta : 1 \to F : \mathcal{C}^{cp} \to \mathcal{C}^{cp}$ be a pointed continuous endofunctor of $\mathcal{C}^{cp}$. Then $\mathsf{Y}\phi \in \mathsf{V}$ names the (underlying type of the) initial F-algebra.

Proof

[a] Immediate from the diagram in §5.7.2.

[b] The comparison $1 \to F1$ is unique and necessarily the image of $\perp \leq \phi\perp$. Repeatedly applying F and ϕ respectively, the Tarski diagram $\perp \leq \phi\perp \leq ... \leq \mathsf{Y}\phi$ is mapped to that in §2.2.3. □

This achieves the unification of §2.1 and §2.2 which we promised.

$^{+}$§**5.7.4** There was an oddity which arose in the $P\omega$ model (§1.4.6) and also in §2.6.7, namely the "raised sum". Can we provide a categorical explanation of this nonassociative operation which seems to be the obvious semantics for tagged records?

Proposition Let $\mathcal{C}$ be a small category of domains containing $\mathcal{T}$ and X, Y two objects of it. Then $X +_{\perp} Y$ is the indexed sum over $\mathcal{T} \to 1$ of the type-dependence which takes the two maximal points of $\mathcal{T}$ to X and Y and $\perp$ to 1. If $f : X \to Z$ and $g : Y \to Z$ are any two continuous functions, there is a *least* continuous function $X +_{\perp} Y \to Z$ making the diagram commute. □

$^{+}$§**5.7.5** What do indexed products over $\mathcal{T} \to 1$ look like?

If we have $\perp, 0, 1 \in \mathcal{T}$ mapped to A, X, Y and the two comparisons are $\alpha : A \to X$ and $\beta : A \to Y$, then the elements of the indexed product are triples $\langle a, x, y\rangle \in A \times X \times Y$ such that $a \leq \mathtt{P}\alpha x, \mathtt{P}\beta y$. Thus in the case $A = 1$ we have a binary product; otherwise we have a new construction: it isn't the product, the pullback or the lax pullback.

$^{+}$§**5.7.6** Now let us look at polymorphism of the "dependent-type" kind, using indexed domain theory. In this case we have variables of "ordinary" (*i.e.* not type) type and quantification over them.

Constant types are interpreted as appropriate 1-indexed domains, $T : 1 \to \mathcal{C}^{cp}$, and *constant values* as (global) elements of these.

Dependent types with variables of type(s) A are interpreted as A-indexed domains, $X(-) : A \to \mathcal{C}^{cp}$. Proliferation of variables is a special case of *substitution*, which is interpreted by precomposition of indexed domains or pullback of fibred ones.

Iterated dependence, *i.e.* with a variable whose type is itself dependent, is interpreted by means of composites of fibred types.

Products, *function spaces* and other categorical constructions are interpreted by means of the appropriate constructions, and *lambda abstraction* is performed using the indexed adjunctive correspondence for the exponential.

Quantification is always over a variable running over a dependent type, which corresponds to a display map; accordingly it is interpreted by the appropriate adjoint to substitution and satisfies the Beck condition as required.

+§**5.7.7** Before looking at the second order polymorphic lambda calculus (which, as we have remarked, we may do in two ways), let us interpret Cardelli's language with V. The differences lie in the new type V and in the interpretation of constructors like $\times$.

The *type-of-types combinator*, V, is interpreted as $\mathbf{t}T : \mathsf{V} \to \mathcal{C}^{cp}$ or as the corresponding $\mathsf{G} \to \mathsf{V}$.

For $\times$, *etc.*, we choose maps $\mathsf{V} \times \mathsf{V} \to \mathsf{V}$ as in proposition 5.7.2; the interpretation of the *expression* $A \times B$ *quâ* element of V now yields the same type as the *construction* (this was the content of the lemma).

Quantification over type-variables is a special case of quantification over any variable, the type of the variable now being V.

+§**5.7.8** Specialising Cardelli's language to the second order lambda calculus, what values to we get for $\forall X.X$, *etc.*? Unfortunately, nothing like what we might expect.

Let $\phi : \mathsf{V} \to \mathsf{V}$ correspond to $\Phi : \mathcal{C}^{cp} \to \mathcal{C}^{cp}$. Then $\forall X.\phi X$ is consists of the assignments to each (name of a) type $\ulcorner X \urcorner$, a point $u\ulcorner X \urcorner \in \Phi X$ such that whenever we have a comparison $f : \ulcorner X \urcorner \leq \ulcorner Y \urcorner$, then $\Phi f(u\ulcorner X \urcorner) \leq u\ulcorner Y \urcorner$, or equivalently $u\ulcorner X \urcorner \leq \Phi g(u\ulcorner Y \urcorner)$ where $f \dashv g$.

Unfortunately there are insufficient such comparisons, so that $\forall X.X$ becomes quite a large product. The point is that we have nothing to force u to preserve isomorphisms, because the whole object of the constructions in the previous section was to dispose of them in order to make V a poset. There will always be lots of distinct elements of V corresponding to isomorphic domains, and we can separate them by open sets (or maps to 2 with different values at two such points).

+§**5.7.9** This brings us back to the "direct" application of indexed domain theory to the second order polymorphic lambda calculus. We interpret a type which depends on a type-variable as a $\mathcal{C}^{cp}$-indexed domain. Since we do not have an explicit *type* of types in this language, it does not matter that $\mathcal{C}^{cp}$ is not itself an object of $\mathcal{C}$. This change is equivalent to the condition that $u \in \forall X.\Phi(X)$ preserve isomorphisms, or that we have the compatibility condition for all comparisons between *domains* rather than *names* for them in V.

We shall intend quantification over type variables to be over $\mathcal{C}^{cp}$ rather than V in future.

Proposition The value of $\forall X.X$ is

(a) the singleton $(\mathsf{K}\bot)$ if $\mathcal{T} \in \mathcal{C}$.

(b) $2 = \{\mathsf{K}\bot, \mathsf{K}\top\}$ otherwise, *i.e.* if every domain in $\mathcal{C}$ has $\top$.

The value of $\forall X.X \to X$ is

(c) the three point lattice if every domain has $\top$.

Proof

[a] Suppose $uX \neq \bot_X$ for some X. We have two embeddings $X \rightarrowtail X \oplus X$, where $X \oplus X$ is the disjoint union of two copies of X with their bottoms identified (this is a retract of $X +_\bot X$ and hence in $\mathcal{C}$ by lemma 2.6.7b). Let the images of uX under these be a, b; then $a, b \leq u(X \oplus X)$. However $\{a, b\}$ is unbounded.

[b] In lemma 2.2.15 we classified the comparisons between X and 2. Suppose $uX \neq \bot_X$ for some X; let $f : X \to 2$ be the characteristic function of $X \setminus \bot$, which is a comparison. Then $f(uX) = \top \leq u2$. Suppose $uY \neq \top$ for some Y; let $y \in Y_{fp} \setminus \downarrow(uY)$ and $g : 2 \to Y$ by $\bot \mapsto \bot$ and $\top \mapsto y$. Then $y = g(\top) \leq uY$, contrary to the choice of y. Hence $u = \mathsf{K}\bot$ or $u = \mathsf{K}\top$.

[c] By considering the comparison $[\bot] : X \to 2$ we see that $\bot \neq u_X \Rightarrow 1 \leq u_2$. By considering the comparison $[x \Rightarrow \top] : 2 \to X$ we see that the latter implies $1 \leq u_X$ and hence (contrapositively) $1 \not\leq u_X \Rightarrow 1 \not\leq u_2 \iff u_2 = \top$. Finally by considering $[\top \Rightarrow \top] : 2 \to X$ we see that this implies $u_X = \top$. Hence uX can only take one of the (uniform) values $\mathsf{K}\bot$, I or $\mathsf{K}\top$. ☐

5.8 General Theory of Typoses

+§**5.8.1** This final section is unashamèdly speculative: it consists of various ideas which, frankly, haven't even really made it to the "top of my head".

The word "typos" was of course a joke, due to Andrew Pitts, the inventor of the tripos. (This is an acronym for "Topos-Representing Indexed Pre-Ordered Set" and was actually coined by Peter Johnstone; I am pleased to say I have never been offered "tripoi" for its plural: even if it *were* a Greek word, the plural would be τριποδες.) However it was clearly the kind of joke that would stick.

I would ask, though, that future workers in this area refrain from using this word, especially in any fixed meaning, until it has become clear what the definition should be, *i.e.* what the strongest possible/reasonable conditions are which can be placed on it. For already in §5.5 above, the term has undergone *two* changes of meaning (besides the reformulations which were the subject of the first part of that section), namely in dropping the (inconsistent) *equality* requirement, and in adding the condition of *extensionality* (§5.5.6). It is likely that when the classical theory developed in this work is extended to domains which are categories rather than posets, and when the ideas are applied to the Effective Topos, and when we begin to see how typoses and toposes can be brought together, then further modification of the definition will be needed, and it would be a pity to lose the use of this word (frivolous though it may have been).

§**5.8.2** This brings us to my embarrassing ignorance of the Effective Topos. Rosolini [1986] discussed the category of domains in this, and it seems likely that if we take the category of comparisons in this then we shall obtain a domain and hence an extensional typos.

§**5.8.3** In proposition 5.5.4 we observed that we only need the (generic fibred type) $\mathsf{G} \to \mathsf{V}$ to determine the typos structure in a category $\mathcal{C}$. Hence so long as we hang on to this map, we can perform constructions on $\mathcal{C}$. In particular the obvious thing to do is to embed it in a topos.

Definition An *internal typos* in a relatively cartesian closed category $\mathcal{E}$ is a morphism $\mathsf{G} \to \mathsf{V}$ satisfying the results of proposition and lemma 5.5.4, *with the exception of axiom (iii)*. An object A of $\mathcal{E}$ is a *domain* if $A \to 1$ can be expressed as a pullback of $\mathsf{G} \to \mathsf{V}$.

If, as is intended, $\mathcal{E}$ is a topos, then it will have lots of other objects which are not domains.

Example Let $\mathcal{C}$ be a category of domains with generic fibred type $\mathsf{G} \to \mathsf{V}$. Put $\mathcal{E} = [\mathcal{C}^{op}, \mathcal{S}]$ and embed $\mathcal{C}$ by the Yoneda embedding. Then we have an internal typos in a topos.

The point of doing this is that whereas we remain within the category of domains when discussing the program-structure and *internal* logic of denotational semantics, we need the full strength of higher-order logic (in the form of topos theory) to discuss the *external* logic. This is needed because there are examples of true theorems which are not provable, or valid terminating programs which are not provably so, in the language and fragment of logic in which they are expressed.

We have shown that categories of domains are complete with respect to products and coproducts over domains; the same work immediately generalises to families of domains indexed over sets (objects of the ambient topos) which are not necessarily domains. Indeed we should think of domains as simply a special kind of set over which the results of programs may range; we may need more general sets in the meta-discussion of the program.

+§**5.8.4** This may throw some light upon the difference between domains with and without $\perp$, and why we do not require it to be preserved (contrary to what usually occurs in Algebra).

We require $\perp$ in domains as the "totally undefined" element and as the zeroth approximation to a fixpoint.

On the other hand, for topos-theoretic purposes it may be useful to have A-indexed families of domains for arbitrary $A \in \mathcal{E}$, not necessarily a domain. This does not conflict with the constructions of the present chapter.

Viewed in a category of posets, A is an arbitrary object (set, poset or whatever) and an A-indexed family of domains is a fibred type (in particular a projection $X \to A$). We may construct products, exponentials, bilimits and fixpoints indexedly over A. Indeed we found fixpoints in proposition 2.4.8.

The formulation of Smyth's theorem about cartesian closure in **AlgPos**, which was the subject of §§2.4.11-13, is now more properly made in this context than simply in terms of products and exponentials in $\mathbf{BiPos}_{\mathrm{f}}$. In other words we want some pullbacks against display maps.

+§**5.8.5** Having now reached the stage of embedding a typos (and with it the model of the λ-calculus and its category of retracts which we used to construct it) in a topos, we are in a position to ask

Question What do we mean by the *spectrum* of a model of the λ-calculus, and what is a *generic* model?

The question of meaning is far from clear: we are asking about a concept defined in terms of function-spaces, and Geometric Logic, the strongest form for which we have spectra or generic models, offers nothing on this topic. Moreover the function-space is of course contravariant in its first argument.

As to some sort of construction, perhaps we should use realisability triposes. How or why, I do not know.

+§**5.8.6** Further on this matter of the spectrum of a model, the object $\forall X.X \to X$ ought to have only two elements.

Question Can $\forall X.X \to X$, or something similar, be given a Heyting algebra structure?

It would appear that the value of $\forall X.X \to X$, as calculated in lemma 5.7.9, depends on the category of domains; it may in fact reflect the hierarchy (§2.6.5) below it.

+§**5.8.7** Discussion of the spectrum of a model brings us to a question which has been avoided in this work just as much as it seems to have been in all others.

Question What is a morphism of models of the λ-calculus?

It would seem reasonable to suppose that a morphism of categories of domains is a functor preserving bilimits, products and exponentials; such a functor preserves λ-models, so (since we would want **Retr** to be functorial) we have to say what is a morphism between models with the same category of retracts.

$^{+}$§**5.8.8** **Retr**(Λ) is not the whole story as to the world in which a model of the λ-calculus naturally resides: we can still vary the base topos (**Set**), though we have not done so at all in this work. Question 5.8.5 should be interpreted to be asking for the natural topos in which Λ resides, and I believe it would be missing the point to suggest that this is **Set** for Λ, $P\omega$ or any of the other models discussed in this work.

The classical name for varying the base topos is *forcing*, and the earliest example of the use of that technique was the independence of the Continuum Hypothesis. It is an easy matter to muck up cardinalities by forcing (consider the classifying topos for the theory of a surjection $\mathbb{N} \twoheadrightarrow X$ for any set X). This is the reason why varying the weight of a category of domains (§2.5.2, §2.6.6 and §2.6.12) is unlikely to make much logical difference.

$^{+}$§**5.8.9** The move to internal typoses in §5.8.3, together with this discussion of morphisms of models, leads to the consideration of subtypos structures on the same category.

Question Let $\mathcal{C} \subset \mathcal{D}$ be the categories of domains for two internal typoses in a topos $\mathcal{S}$, so products and exponentials in $\mathcal{C}$ are inherited from $\mathcal{D}$ and $\mathcal{S}$. Does the inclusion have a left adjoint?

Bibliography

S. Abramsky

[1986] *Domain Theory in Logical Form*, 1986

P. Aczel

[1980] *Frege Structures*, in [Barwise *et al.* 1980]

J.F. Adams
A.V. Aho and J.D. Ullman

[1977] *Principles of Compiler Design*, Addison-Wesley, 1977

ANSI

[1966] *American National Standard* `FORTRAN` **X.3.9** (1966), American National Standards Institute, New York

M.A. Arbib
M.A. Arbib and E.G. Manes

[1975] *Arrows, Structures and Functors — the Categorical Imperative*, Academic Press, 1975, `MR 51/638`

M. Artin, A. Grothendieck and J.L. Verdier

[1964] *Théorie des Topos et cohomologie etale des schèmas*, Séminaire de Géometrie Algebraique du Bois Marie 1963/4, LNM **269 & 270** (1964) 1-340, Springer, `MR 50/7130-1`

E. Astesiano and C. Böhm

[1981] *Trees in algebra and programming* (Genoa 1981), 1981

G. Ausiello and C. Böhm

[1979] *Automata, languages and programming* (Udine 1978), 1979

C.A. Babbage
? Baclawski

[] *Homology of Posets*, Adv Maths,

J. Baeten and B. Boerboom

[1979] Ω *can be anything it shouldn't be*, Nederl. Akad. Wetensch. Indag. Math. **41** (1979) 111-120, `MR 80h:03018`

B. Banaschewski

[1982] *Categorical Aspects of Topology and Analysis* (Ottawa 1981), LNM **915** (1982), Springer

B. Banaschewski and R.-E. Hoffman

[1981] *Continuous Lattices* (Bremen 1979), LNM **871** (1981), Springer

H.P. Barendregt

[1975] *λ-calculus and Computer Science Theory* (Rome 1975), LNCS **37** (1975), Springer

[1981] *The Lambda Calculus — its Syntax and Semantics*, SLFM **103** (1981), North-Holland, MR 83b:03016

H.P. Barendregt, M. Coppo and M. Dezani-Ciancaglini

[1983] *A filter lambda model and the completeness of type assignment*, J. Symbolic Logic **48/4** (1983) 931-940

H.P. Barendregt and M. Dezani-Ciancaglini

[1982] *International Symposium on Programming* (Turin 1982), 1982

H.P. Barendregt and G. Longo

[1980] *Equality of Lambda Terms in the Model $\mathcal{T}$*, in [Hindley & Seldin 1980] 303-337, MR 82g:03017

M. Barr

[1969] *Midwest Category Seminar*, LNM **106** (1969), Springer

[1971] *Exact Categories*, Exact Categories and Categories of Sheaves, LNM **236** (1971) 1-120, Springer, Z 223:18010

[1979] **-autonomous categories*, 1979

M. Barr and S. Mac Lane

[1970] *Reports of the Midwest category seminar 3*, 1970

M. Barr and C. Wells

[1985] *Toposes, Triples and Theories*, Gr. d. Math. Wis. **278** (1985), Springer

J. Barwise

[1975] *Admissible sets and structures : an approach to definability theory*, 1975

[1977] *Handbook of Mathematical Logic*, SLFM **90** (1977), North-Holland North-Holland SLFM

J. Barwise, A. Baudisch and S. Feferman

[1983] *Model-theoretic logics*, 1983

J. Barwise, H.J. Keisler and K. Kunen

[1980] *The Kleene Symposium* (Madison, 1978), SLFM **101** (1980), North Holland

A. Baudisch see J. Barwise
M.J. Beeson

[1981] *Formalising Constructive Mathematics: why and how?*, in [Richman 1981] 146-190

[1985] *Foundations of Constructive Mathematics*, Springer, 1985

J. Bénabou

[1975] *Fibrations Pétites et Localement Pétites*, Comptes Rendues Acad. Sci. Math. **281** (1975) 831-834 & 897-900, Paris, MR 52/13991

[1985] *Fibred Categories and the Foundations of Naïve Category Theory*, J. Symb. Logic **50** (1985) 10-37

G. Birkhoff and S. Mac Lane

[1967] *Algebra*, Macmillan, 1967

B. Boerboom see J. Baeten
C. Böhm see also E. Astesiano and G. Ausiello2.5.4

[1975] *λ-calculus and Computer Science Theory* (Rome 1975), LNCS **37** (1975), Springer

A. Boileau and A. Joyal

[1981] *La Logique des Topos*, JSL **46** (1981) 6-16

F. Borceux and G. Van den Bossche

[1983] *Algebra in a localic topos with applications to ring theory*, LNM **1038** (1983), Springer

L.E.J. Brouwer and D.S. van Dalen

[] *Brouwer's Cambridge lectures on intuitionism*,

R. Brown

[1964] *Function Spaces and Product Topologies*, Q. J. Math., 1964

K.B. Bruce and A.R. Meyer

[1984] *The Semantics of Second Order Polymorphic Lambda Calculus*, in [Kahn *et al.* 1984] 131-144

G. Cantor
L. Cardelli

[] *Two-dimensional syntax for functional languages*,

[1986] *A Polymorphic Lambda Calculus with* **Type:Type**, Digital Equipment Corporation, 1986

J. Celeyrette

[1975] *Fibrations et Extensions de Kan*, Thèse, Univ de Paris-Nord, 1975

C.C. Chang and H.J. Keisler

[1973] *Model Theory*, North-Holland, 1973

J.M. Chapman

[1986] *Relative Category Theory as an approach to Geometric Morphisms*, Ph.D. dissertation, Bristol, 1986

A. Church

[1941] *The Calculi of Lambda Conversion*, Princeton University Press, 1941

[1953] *Introduction to mathematical logic*, 1953

[1976] *A System of Set Theory with a Universal Set*, Tarski Symposium, American Mathematical Society, 1976

T.W. Clarke, P.J.S. Gladstone, C.D. Maclean and A.C. Norman

[1980] *SKIM - the* S, K, I *Reduction Machine*, LISP-80, 1980

W.F. Clocksin
W.F. Clocksin and C.S. Mellish

[1981] *Programming in* `PROLOG`, Springer, 1981

P.M. Cohn

[1965] *Universal Algebra*, Harper and Row, 1965

M. Coppo see H.P. Barendregt
Th. Coquand, C.A. Gunter and G. Winskel

[1986] *Polymorphism and domain equations*, 1986

M. Coste

[1980] *Localisation, Spectra and Sheaf Representation*, in [Fourman *et al.* 1980] 212-238

G. Cousineau, P.-L. Curien and B. Robinet

[1986] *Categorical combinators, sequential algorithms and functional programming*, Combinators and functional programming languages (Val d'Ajol 1985), 1986

P.-L. Curien see also G. Cousineau

[1983] *Combinateurs Catégoriques, algorithmes séquentiels et Programmation Applicative*, Ph.D. dissertatation, Paris VII, 1983

P.-L. Curien and A. Obtułowicz

[] *Partiality and Cartesian Closedness*,

H.B. Curry

[1942] *The Inconsistency of Certain Formal Logics*, J. Symb. Log. **7** (1942) 115-117

[1963] *Foundations of mathematical logic*, 1963

H.B. Curry and R. Feys

[1958] *Combinatory Logic I*, North-Holland, 1958

H.B. Curry, J.R. Hindley and J.P. Seldin

[1972] *Combinatory Logic II*, North-Holland, 1972

A. Day

[1975] *Filter Monads, Continuous Lattices and Closure Systems*, Canad. J. Math. **27** (1975) 50-59, `MR 51/3258`

M. Dezani-Ciancaglini see H.P. Barendregt
S. Eilenberg

[] *Automata, Languages and Machines*,

S. Eilenberg and C.C. Elgot

[] *Recursiveness*,

S. Eilenberg, A. Heller and M. Tierney

[1975] *Algebra, Topology and Catgeory Theory: a collection of papers in honor of Sammy Eilenberg*, 1975

S. Eilenberg and S. Mac Lane

[1942] *Natural Isomorphisms in Group Theory*, Proc. Nat. Acad. Sci. USA **28** (1942) 537-543, `MR 4-134`

C.C. Elgot see S. Eilenberg
Y. Ershov
J. Fairbairn

[1984] *A New Type-Checker for a Functional Language.* **53** (1984), Univ. Camb. Comp. Lab. techn. report

J. Fairbairn and S. Wray

[1987] `TIM`*: A Simple, lazy Abstract Machine to Execute Supercombinators*, Dept of Computer Science Research Report **87/R6** (1987), Univ. Glasgow

S. Feferman see J. Barwise
R. Feys see H.B. Curry
R.G. Fisker, C.H.A. Koster, C.H. Lindsey, B.J. Mailloux, J.G.L.T. Meertens, J.L. Peck, M. Sintzoff and A. van Wijngaarden

[1975] *The Revised Report on the Algorithmic Programming Language* `ALGOL 68`, Acta Informatica **5:1-3** (1975) 1-236

T.E. Forster

[1983] *Quine's New Foundations: an Introduction*, Cahiers du Centre de Logique, 1983

M.P. Fourman

[1974] *Connections between Category Theory and Logic*, D.Phil. dissertation, Oxford, 1974

[1977] *The Logic of Topoi*, in [Barwise 1977] 1053-1090, `MR 56/15351 58/10343`

M.P. Fourman, C.J. Mulvey and D.S. Scott

[1979] *Applications of Sheaves* (Durham, 1977), LNM **753** (1979), Springer

[1980] *Applications of Sheaves*, 1980

M.P. Fourman and D.S. Scott

[1979] *Sheaves and Logic*, in [Fourman *et al.* 1979] 302-401, `MR 82d:03061`

M.P. Fourman and S.J. Vickers

[1986] *Theories as Categories*, in [Pitt 1986] 434-448

G. Frege
P.J. Freyd

[] *Numerals and Ordinals*,

P.J. Freyd and G.M. Kelly

[1972] *Categories of Continuous Functors*, J Pure Appl Alg **2** (1972) 169-191, `MR 48/369`

E G. Kahn, D.B. MacQueen and G.D. Plotkin

[1984] *Semantics of Data Types*, LNCS **173** (1984), Springer

B.D. Gabbay

[1984] *Elementary Logic — A Procedural Perspective*, 1984

P. Gabriel and F. Ulmer

[1971] *Lokal Präsentierbare Kategorien*, LNM **221** (1971), Springer, MR 48/6205

G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove and D.S. Scott

[1980] *A Compendium of Continuous Lattices*, Springer, 1980, MR 82h:06005

J.-Y. Girard

[1972] *Interprétation functionelle et élimination des coupures de l'arithmétique d'ordre superieur*, Ph.D. dissertation, Paris VII, 1972

[1985] *The System* F *of Variable Types, Fifteen Years Later*, 1985

J. Giraud

[1972] *Classifying Topos*, in [Lawvere 1972] 43-56, MR 50/2300

P.J.S. Gladstone see T.W. Clarke
R.I. Goldblatt

[1979] *Topoi: the Categorical Analysis of Logic*, SLFM **98** (1979), North-Holland, MR 81a:03063

A. Grothendieck see M. Artin
A.J. Guilfoyle

[1982] *Computable Analysis*, 1982

C.A. Gunter see also Th. Coquand

[1985] *Profinite Solutions of Recursive Domain Equations*, Ph.D. dissertation, Wisconsin-Addison, 1985

[1987] *Universal Profinite Domains*, Information and Computation **72** (1987) 1-30

C.A. Gunter and D.S. Scott

[1987] *Semaintic Domains*, 1987

A. Heller see also S. Eilenberg

[1970] *Applications of Categorical Algebra* (New York), American Mathematical Society, 1970

L. Henkin and A. Tarski

[1974] *The Tarski Symposium* (California, 1971), 1974

J.R. Hindley see also H.B. Curry

[1983] *The Completeness Theorem for Typing λ-terms*, Theor. Comp. Sci. **22** (1983) 1-17 & 127-133

J.R. Hindley, B. Lercher and J.P. Seldin

[1972] *Introduction to Combinatory Logic*, LMS Lecture Notes **7** (1972), CUP

J.R. Hindley and J.P. Seldin

[] *Introduction to combinators and the λ-calculus*,

[1979] *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, 1979

[1980] *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism To H.B. Curry: essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, 1980

C.A.R. Hoare and N. Wirth

[1973] *An Axiomatic Definition of the Programming Language* `PASCAL`, Acta Informatica **2:4** (1973) 335-356

R.-E. Hoffman see B. Banaschewski
K.H. Hofmann see G. Gierz
Ch. Hosono and M. Sato

[1977] *The Retracts in $P\omega$ do not form a Continuous Lattice*, Theor. Comp. Sci. **4** (1977) 137-142, `MR 58/8442`

W.A. Howard

[1980] *The formulae-as-types notion of construction*, in [Hindley & Seldin 1980] 479-490

J.M.E. Hyland

[1975] *A Survey of some useful Partial Order Relations on Terms of the Lambda Calculus*, in [Barendregt 1975] 83-95, `MR 57/9496`

[1976] *A Syntactic Characterisation of Equality in some Models of the λ-calculus*, J. LMS **II 12** (1976) 361-370, `MR 53/100`

[1981] *Function Spaces in the Category of Locales*, in [Banaschewski & Hoffman 1981] 264-281

[1982] *The Effective Topos*, in [Troelstra & van Dalen 1982] 165-216, `MR 84m:03101`

J.M.E. Hyland, P.T. Johnstone and A.M. Pitts

[1980] *Tripos Theory*, Math. Proc. Camb. Philos. Soc. **88** (1980) 205-232, `MR 81i:03102`

C.B. Jay

[1987] *A Language for Monoidal Categories*, 1987

K. Jensen and N. Wirth

[1975] `PASCAL` *User Manual and Report*, Springer, 1975

P.T. Johnstone see also J.M.E. Hyland1 4.4.5 5.6.1

[1977] *Topos Theory*, LMS Monograph **10** (1977), Academic Press, `MR 57/9791`

[1980] *Open Maps of Toposes*, Manuscripta Math. **31** (1980) 217-247, `MR 81f:18020`

[1981] *Factorisation Theorems for Geometric Morphisms I*, Cah. de Top. et Géom. Diff. **22** (1981) 33-47, `MR 82j:18007`

[1981] *Scott is not always sober*, in [Banaschewski & Hoffman 1981] 282-283

[1982] *The Vietoris Monad on the Category of Locales*, Math. Arb. pap. **27** (1982) 162-179, Univ. Bremen

[1982] *Factorisation Theorems for Geometric Morphisms II*, in [Banaschewski 1982] 216-233, `MR 83j:18006`

[1983] *Fibred Categories*, St. Adv. Maths. **3** (1983), Part III Lecture Course, Cambridge, `MR 85f:54002`

[1983] *Open Locales and Exponentiation*, Mathematical Applications of Category Theory (Denver 1983), AMS, 1983, `MR 85i:18005`

P.T. Johnstone and A. Joyal

[1982] *Continuous Categories and Exponentiable Toposes*, J. Pure Appl. Alg. **25** (1982) 255-296, `MR 83k:18005`

P.T. Johnstone, R. Paré, R.D. Roseburgh, D. Schumacher, R.D. Wood and G.C. Wraith

[1978] *Indexed Categories and their Applications*, LNM **661** (1978), Springer, `MR 58/16816`

A. Joyal see A. Boileau and P.T. Johnstone
A. Joyal and M. Tierney

[] *An Extension of the Galois Theory of Grothendieck*,

G. Kahn, D.B. MacQueen and G.D. Plotkin

[1984] *Semantics of Data Types* (Sophia-Antipolis, 1984), LNCS **173** (1984), Springer

K. Keimel see G. Gierz
H.J. Keisler see J. Barwise and C.C. Chang
G.M. Kelly see also P.J. Freyd

[1982] *Basic Concepts of Enriched Category Theory*, LMS lecture notes **64** (1982), CUP

B.W. Kernighan, M.E. Lesk and D.M. Richie

[1975] *The* `C` *Programming Language*, CSTR **31** (1975), Bell Labs, N.J.

S.C. Kleene and J.B. Rosser

[1935] *The Inconsistency of Certain Formal Logics*, Ann. Math. **36** (1935) 630-636

D. Knuth

[1968] *The Art of Computer Programming*, Addison-Wesley, 1968

A. Kock

[1981] *Synthetic differential geometry*, LMS lecture notes **51** (1981), CUP

A. Kock and G.E. Reyes

[1977] *Doctrines in Categorical Logic*, in [Barwise 1977] 283-313, `MR 58/10395`

A. Kock and G.C. Wraith

[1971] *Elementary Toposes*, Lecture Notes **30** (1971), Aarhus Universitet, `MR 49/7324`

C.H.A. Koster see R.G. Fisker
C.P.J. Koymans

[1984] *Models of the Lambda Calculus* **9** (1984), Centrum voor Wiskunde en Informatica, Amsterdam, `?  MR 85b:03020`

K. Kunen see J. Barwise
Y.G.A. Lafont

[1987] *From Category Theory to Implementation*, 1987

[1987] *The Linear Abstract Machine*, 1987

J. Lambek

[1979] *From λ-calculus to Cartesian Closed Categories*, in [Hindley & Seldin 1979] 375-402

[1980] *From Types to Sets*, Advances in Mathematics **35** (1980)

J. Lambek and P.J. Scott

[1986] *An Introduction to Higher Order Categorical Logic*, St. Adv. Maths. **7** (1986), CUP

K.G. Larsen

[1987] *Proof Systems for Hennessy-Milner Logic with Recursion*, 1987

J.D. Lawson see G. Gierz
F.W. Lawvere

[1963] *Functorial Semantics of Algebraic Theories*, Proc Nat Acad Sci USA **50** (1963) 869-872, MR 28/2143

[1964] *An Elementary Theory of the Category of Sets*, Proc. Nat. Acad. Sci. **52** (1964) 1506-1511

[1969] *Diagonal Arguments and Cartesian Closed Categories*, Category Theory, Homology Theory and their Applications II, LNM **92** (1969) 134-145, Springer

[1969] *Adjointness in Foundations*, Dialectica **23** (1969) 281-296

[1970] *Quantifiers and Sheaves*, Actes du Congrès Intern. des Math., Nice **1** (1970) 329-334

[1972] *Toposes, Algebraic Geometry and Logic* (Dalhousie, 1971), LNM **274** (1972), Springer

[1973] *Metric Spaces, Generalised Logic and Closed Categories*, 1973

F.W. Lawvere, C. Maurer and G.C. Wraith

[1975] *Model theory and topoi : a collection of lectures by various authors*, 1975

B. Lercher see J.R. Hindley
M.E. Lesk see B.W. Kernighan
L.G. Lewis

[1983] *Open maps, colimits and a convenient category of fibre spaces*, 1983

T. Lindgren

[1984] *Untitled manuscript*, Ph.D. dissertation, Rutgers University, 1984

C.H. Lindsey see R.G. Fisker
F.J. Linton
J.W. Lloyd

[1984] *Foundations of Logic Programming*, Technical Report **82/7** (1984), Dept of Comp Sci, Univ Melbourne

G. Longo see H.P. Barendregt
G. Longo and E. Moggi

[1984] *Cartesian closed categories and partial morphisms for effective type structures The Semantics of Second Order Polymorphic Lambda Calculus*, in [G. *et al.* 1984] 131-144

[1984] *The hereditary partial functional and recursion theory in higher types*, JSL **4/49** (1984)

S. Mac Lane see also M. Barr, G. Birkhoff and S. Eilenberg1.1 1.1.1 1.2.13 3.1.2 4.2.14 4.4.2

[1971] *Categories for the Working Mathematician*, Springer, 1971

C.D. Maclean see T.W. Clarke
D.B. MacQueen see E G. Kahn and G. Kahn
D.B. MacQueen and D.T. Sanella

[] *Completeness of proof systems for equational specifications*,

D.B. MacQueen and R. Sethi

[1982] *A Semantic Model of Types for Applicative Languages*, Symposium on LISP and Functional Programming, 1982

[1982] *A Higher-Order Polymorphic Type System for Applicative Languages*, Symposium on LISP and Functional Programming (1982) 242-252

[1984] *An Ideal Model for Recursive Polymorphic Types*, Eleventh Annual Symposium on Principles of Programming Languages, 1984

B.J. Mailloux see R.G. Fisker
M. Makkai and G.E. Reyes

[1977] *First Order Categorical Logic: Model-theoretic methods in the theory of topoi and related categories*, LNM **611** (1977), Springer, `MR 58/21600`

E.G. Manes see also M.A. Arbib

[1975] *Algebraic Theories*, GTM **26** (1975), Springer , `MR 54/7578`

P. Martin-Löf

[1971] *A Theory of Types*, unpublished, 1971

[1975] *An Intuitionistic Theory of Types: Predicative Part*, Logic Colloquium, North-Holland, 1975

[1980] *Intuitionistic Type Theory*, Bibliopolis, 1980

A.R.D. Mathias

[1979] *Surrealist landscape with figures (a survey of some recent results in set theory)*, Per Math Hungarica **10** (1979) 109-175

[1987] *The Ignorance of Bourbaki*, Eureka **46** (1987)

C. Maurer see F.W. Lawvere
A.D. McGettrick

[1978] *Algol 68: A First and Second Course*, Computer Science Text, CUP, 1978

J.G.L.T. Meertens see R.G. Fisker
C.S. Mellish see W.F. Clocksin
A.R. Meyer see also K.B. Bruce

[1982] *What is a Model of the Lambda Calculus?*, Information and Control **52** (1982) 87-122

R. Milner

[1978] *A Theory of Type Polymorphism in Programming*, J Comp Sys Sci **17** (1978) 348-375

[1982] *Calculi for Synchrony and Asynchrony*, 1982

M. Mislove see G. Gierz
B. Mitchell

[1965] *Theory of Categories*, Academic Press, 1965

J.C. Mitchell and E. Moggi

[1987] *Kripke-style models for typed lambda calculus*, 1987

E. Moggi see also G. Longo and J.C. Mitchell

[1986] *Polymorphism and Internal Categories*, Peripatetic Seminar on Sheaves and Logic (Cambridge, March 1986), 1986

[1986] *Partial Morphisms in Categories of Effective Objects*, 1986

[1987] *Empty Types in Polymorphic Lambda Calculus*, 1987

G.P. Monro

[1984] *A Category-theoretic approach to Boolean-valued models*, 1984

C.J. Mulvey see M.P. Fourman
R. Nakajima

[1975] *Infinite Normal Forms for the λ-calculus*, in [Böhm 1975] 62-82

P. Naur

[1963] *Revised Report on the Algorithmic Programming Language ALGOL 60*, Comm. ACM. **6:1** (1963) 1-17

S.B. Niefield

[1982] *Exactness and Projectivity*, 1982

S.B. Niefield and K.I. Rosenthal

[1986] *Constructing locales from quantales*, 1986

M. Nielsen and E.M. Schmidt

[1982] *Automata, Languages and Programming IX Automata, Languages and Programming, Aarhus 1982* (Aarhus, 1982), LNCS **140** (1982), Springer

A.C. Norman see T.W. Clarke
A. Obtułowicz see also P.-L. Curien

[1986] *categorical and algebraic aspects of Martin-Löf type theory*, 1986

[1987] *Algebra of Constructions I: The Word Problem for Partial Algebras*, Information and Computation **73** (1987) 129-173

A. Obtułowicz and A. Wiweger

[1982] *Algebraic, Functional and Categorical aspects of the type-free Lambda Calculus*, in [Traczyk 1982] 299-324

R. Paré see P.T. Johnstone
D. Park

[1976] *The* Y*-combinator in Scott's Lambda Calculus Models (revised)*, Dept. Comp. Sci. Theory of Computation report **13** (1976), Univ. Warwick

J.L. Peck see R.G. Fisker
J. Penon

[1974] *Catégories localement internes*, C.R. Acad. Sci. Math. **278** (1974) A1577-1580, Paris, MR 51/10435

F.C.N. Pereira and D.H.D. Warren

[1980] *Definite Clause Grammars: A Survey pf the Formalism and a Comparison with Augmented TRansition Networks*, Artif Intell **13** (1980) 231-278

D. Pitt

[1986] *Category Theory and Computer Programming* (Guildford, 1985), LNCS **240** (1986), Springer

A.M. Pitts see also J.M.E. Hyland0 5.6.1

[1981] *The Theory of Triposes*, Ph.D. dissertation, Cambridge, 1981

[1987] *Completeness of topos models for polymorphic lambda calculus*, Categories in computer science and logic (Boulder 1987), 1987

G. Plotkin5.4
G.D. Plotkin see also E G. Kahn and G. Kahn

[1974] *The λ-calculus is ω-incomplete*, J. Symbolic Logic **39** (1974) 313-317, MR 50/6799

[1976] *A Powerdomain Construction*, SIAM J. Comp. **5** (1976) 452-487, MR 56/4224

[1978] *$\mathcal{T}$ as a Universal Domain*, J. Comp. Sys. Sci. **17** (1978) 209-236, MR 80d:68105

[1984] *Types and partial functions*, 1984

G.D. Plotkin and M.B. Smyth

[1978] *The Category Theoretic Solution of Recursive Domain Equations*, Dept. Artif. Intell. research report **60** (1978), Edinburgh Univ.

A. Poigné

[1986] *Foundations are rich Institutions, but Institutions are poor Foundations*, 1986

V.W.V. Quine

[1937] *New Foundations for Mathematical Logic*, American Mathematical Monthly, 1937

F. Ramsey
G.E. Reyes see A. Kock and M. Makkai
J.C. Reynolds

[1984] *Polymorphism is not Set-Theoretic*, LNCS **173** (1984), Springer

D.M. Richie see B.W. Kernighan
F. Richman

[1981] *Constructive Mathematics* (New Mexico 1980) **873** (1981), Springer

B. Robinet see G. Cousineau
E.P. Robinson

[1986] *A modal language for powerdomains*, 1986

R.D. Roseburgh see P.T. Johnstone
K.I. Rosenthal see S.B. Niefield
G. Rosolini

[1985] *Topoi and p-Categories*, Carnegie-Mellon University, 1985

J.B. Rosser see also S.C. Kleene

[1942] *The Burali-Forti Paradox*, J Symb Logic **7** (1942) 1-17

B.A.W. Russell

[1908] *Mathematical Logic based on the Theory of Types*, Amer. J. Math. **30** (1908) 222-263

B.A.W. Russell and A.N. Whitehead

[1910-13] *Principia Mathematica I-III*, CUP, 1910-13

D.T. Sanella see D.B. MacQueen
D. Sannella and A. Tarlecki

[1985] *Specifications in an arbitrary institution*, Internal Report **184** (1985), Univ Edinburgh, Dept of Computer Science

M. Sato see Ch. Hosono
A. Ščedrov

[1984] *Forcing and Classifying Topoi*, Mem. Amer. Math. Soc. **48** (1984)

E.M. Schmidt see M. Nielsen
D. Schumacher see P.T. Johnstone
D.S. Scott see also M.P. Fourman, G. Gierz and C.A. Gunter1.1 1.1.1 1.4.1 1.4.7-8 2.2.1 2.3.5 2.4.1 2.5.11 2.5.3 5.5.1

[1970] *Outline of a mathematical theory of computation*, monograph **2** (1970), Oxford Univ Programming Support Group

[1970] *The lattice of flow diagrams*, monograph **3** (1970), Oxford Univ Programming Support Group

[1972] *Lattice Theory, Data Types and Semantics*, Formal Semantics of Programming Languages, 1970 (1972) 65-106, Prentice-Hall, `MR 56/7304`

[1972] *Continuous Lattices*, in [Lawvere 1972] 97-136, `MR 53/7879`

[1975] *Combinators and classes*, in [Böhm 1975]

[1976] *Data Types as Lattices*, SIAM J. Comp. **5** (1976) 522-587, `MR 55/10262`

[1979] *Identity and Existence in Intuitionistic Logic*, in [Fourman *et al.* 1979] 660-696, `MR 81a:03060`

[1980] *Lambda Calculus: some Models, some Philosophy*, in [Barwise *et al.* 1980] 223-265

[1981] *Lectures on a Mathematical Theory of Computation*, PRG Techn. Monogr. **19** (1981), Oxford Univ., `MR 85g:68043`

[1982] *Domains for Denotational Semantics*, in [Nielsen & Schmidt 1982] 577-613, `MR 83m:68029`

D.S. Scott and C. Strachey

[1981] *Towards a mathematical semantics for computer languages*, monograph **6** (1981), Oxford Univ Programming Support Group

P.J. Scott see J. Lambek
J.A. Seebach and L.A. Steen

[1978] *Counterexamples in Topology*, Hold, Rinehart & Winston, 1978, `MR 42/1040`

R.A.G. Seely

[1977] *Hyperdoctrines and Natural Deduction*, Ph.D. dissertation, Cambridge, 1977

[1983] *Natural Deduction and the Beck Condition*, Zeitschr. Math. Logik und Grundlagen d. Math. **29** (1983) 505-542

[1984] *Locally Cartesian Closed Categories and Type Theory*, Proc. Camb. Philos. Soc. **95** (1984), `MR 83m:03076`

[1985] *Higher Order Polymorphic Lambda Calculus and categories I & II*, Math Reports Canad Acad Sci, 1985

J.P. Seldin see H.B. Curry and J.R. Hindley
R. Sethi see D.B. MacQueen
H. Simmons

[1978] *The lattice-theoretic part of topological separation properties*, Proc Edinb Math Soc **21** (1978) 41-48

M. Sintzoff see R.G. Fisker
M.B. Smyth see also G.D. Plotkin11 5.6.4

[1982] *The Largest Cartesian Closed Category of Domains*, Dept. Sci. Sci. internal report **108** (1982), Edinburgh Univ.

[1983] *Power domains and predicate transformers: a topological view*, Techincal monograph **126** (1983), Univ Edinburgh Dept of Computer Science

[1983] *The Largest Cartesian Closed Category of Domains Theor Comp Sci* **27** (1983) 109-119, Edinburgh Univ.

I. Sols

[1976] *Programming in Topoi*, Cahiers top. et Géom. diff. **16** (1976) 312-319

T.P. Speed

[1972] *Profinite Posets*, Bull Austral Math Soc **6** (1972) 177-183, `MR 45/4371`

L.A. Steen see J.A. Seebach
J.E. Stoy

[1977] *Denotational Semantics: the Scott-Strachey approach to Programming Languages*, MIT Press, Cambridge, Mass., USA, 1977, `MR 58/8460`

C. Strachey see also D.S. Scott

[1973] *The varieties of programming language*, monograph **10** (1973), Oxford Univ PRG

C. Strachey and C.P. Wadsworth

[1974] *Continuations: a mathematical semantics for handling full jumps*, monograph **11** (1974), Oxford Univ PRG

A. Tarlecki see D. Sannella
A. Tarski see also L. Henkin8

[1955] *A Lattice-theoretical Fix-point Theorem and its Applications*, Pacific J. Math. **5** (1955) 285-309, `MR 17-574`

B.R. Tennison
M. Tierney see S. Eilenberg and A. Joyal
T. Traczyk

[1982] *Universal Algebra and Applications* **9** (1982), Banach Center Publications

A.S. Troelstra

[] *Choice sequences : a chapter of intuitionistic mathematics*,

A.S. Troelstra and D.S. van Dalen

[1982] *L.E.J. Brouwer Centenary Symposium* (Noordwijkerhout, 1981) **110** (1982), North-Holland

A.M. Turing
J.D. Ullman see A.V. Aho
F. Ulmer see P. Gabriel
D. van Dalen

[1983] *Logic and Structure*, Universitext, Springer, 1983

D.S. van Dalen see also L.E.J. Brouwer and A.S. Troelstra

[] *Extension problems in intuitionistic plane projective geometry*,

G. Van den Bossche see F. Borceux
A. van Wijngaarden see R.G. Fisker
J.L. Verdier see M. Artin
S.J. Vickers see also M.P. Fourman

[1986] `PROLOG` — *a sheaf-theoretic semantics*, 1986

[1987] *An algorithmic approach to p-adic integers*, 1987

C.P. Wadsworth see C. Strachey
D.H.D. Warren see F.C.N. Pereira
C. Wells see M. Barr
A.N. Whitehead see B.A.W. Russell
G. Winskel see also Th. Coquand

[1984] *Categories of models for concurrency*, 1984

[1985] *A note on powerdomains and modality*, Theor Comp Sci **36** (1985) 127-137

[1985] *A complete proof system for SCCS with modal assertions*, 1985

N. Wirth see C.A.R. Hoare and K. Jensen
A. Wiweger see A. Obtułowicz
R.D. Wood see P.T. Johnstone
G.C. Wraith see P.T. Johnstone, A. Kock and F.W. Lawvere1
S. Wray see J. Fairbairn

Index