From term models to domains

Wesley Phoa
Laboratory for the Foundations of Computer Science
James Clerk Maxwell Building, The King's Buildings
Edinburgh EH9 3JZ, Scotland

Abstract

Let B be the closed term model of the λ-calculus in which terms with the same Böhm tree are identified. We investigate which partial equivalence relations (PERs) on B can be regarded as predomains or domains. Working inside the realizability topos on B, such PERs can be regarded simply as sets in a particular model of constructive set theory.

No well-behaved partial order has been identified for any class of PERs; but it is still possible to isolate those PERs which have 'suprema of chains' in a certain sense, and all maps between such PERs in the model preserve such suprema of chains. One can also define what it means for such a PER to have a 'bottom'; partial function spaces provide an example. For these PERs, fixed points of arbitrary endofunctions exist and are computed by the fixed point combinator y.

The categories of predomains are closed under the formation of total and partial function spaces, polymorphic types and convex powerdomains. (Subtyping and bounded quantification can also be modelled.) They in fact form reflective subcategories of the realizability topos; and in this set-theoretic context, these constructions are very simple to describe.

We illustrate the theory by discussing an interpretation of PCF.

1 Introduction

This paper is an example of a new approach to domain theory: one in which a category of domains is built up directly from a particular model of computation, in a natural and intuitive framework, and using the tools of category theory and topos theory.

Domains are traditionally thought of as complete partial orders, of one kind or another. This makes it possible to interpret recursion, and to solve recursive domain equations; and indeed, classical domain theory has been very successful. But it has its shortcomings: for example,

- The mathematics can get quite complicated, even though the underlying intuitions may be very simple (though admittedly many of the complications arise when domain theory is studied for its intrinsic interest rather than with a view to applications).

- Some programming language features (such as polymorphism and nondeterminism) are difficult to model, especially in combination.

- The classical treatment of effectivity, in terms of enumerated bases, is technically difficult and is not, intensionally, related to the 'computations' (reductions) that go on at an operational level.

- It's hard to establish a relationship with operational semantics.

One problem may be that domain theory throws away the idea of computability and replaces it with a mathematical analogy, namely continuity. It seems worthwhile to look for other models of typed computations.

For a few years now, a different approach towards domain theory has been emerging, one which incorporates computability in a direct and concrete way and which suggests new ways of attacking the problems listed above. Not everyone agrees on what it is, and certainly the results are still poorly understood, but it seems to combine at least two ideas:

- *A domain is a particular kind of set* in some appropriate model of constructive set theory. That is, we don't need to impose a complete partial order or any such additional structure; this is automatic, all set-theoretic functions between domains respect this automatic structure, and we can reason about domains—and perform constructions on domains—in a purely set-theoretic way. For example, exponentials are the full set-theoretic function spaces.

- *A domain is a particular kind of partial equivalence relation* on some model of the λ-calculus; or more generally, on some partial combinatory algebra. That is, we look for classes of domains sitting inside PER models for polymorphism. The fact that our theory is 'effective' is now clear from the fact that we are using PERs and maps between them—enumerations are not used.

(Both are due, in one way or another, to Dana Scott: see [31].) Note that one can regard *computability* as being built into both approaches; rather than singling out 'effective domains' and 'computable functions', *all* maps are computable: from the second point of view by definition, from the first by virtue of the particular model we pick.

The best way to combine the two approaches is to work inside a realizability topos. Roughly speaking, a *topos* is a category-theoretic model of constructive set theory; a *realizability topos* is a topos constructed using a partial combinatory algebra—for example, the partial recursive functions, or a model of the untyped λ-calculus. Such a topos contains a counterpart of the category of PERs, namely the (internal) category of *modest sets*: see [9] and [14].

A lot of effort has been concentrated on looking for domains in the effective topos (the realizability topos based on the partial recursive functions): first came the work of Rosolini [30], and later Phoa [20] and Freyd et al. [8], among others. More recently, people have been looking at continuous models of the λ-calculus: see, for example, the work of Amadio, and of Abadi and Plotkin [1] (these results have not yet been fitted into the topos-theoretic framework).

In this model we look for domains in the realizability topos based on a certain term model of the λ-calculus. Some of the proofs are rather technical, but this does not mean that *using* these unfamiliar domains is more difficult than using, say, Scott domains—in fact, it can be easier (for instance, we never have to bother to check that a function is continuous, as this is automatic). However, the paper is not really concerned with applications, but with the basic theory only.

The reader uncomfortable with topos theory should not be put off. The definitions and proofs in this paper make sense, word for word, as statements about PERs in the usual ('external') sense—though, read in this way, they are of course weaker. The fact that we're actually talking about PERs in an odd constructive universe is somehow transparent. So perhaps, on a first reading, it's safe just to ignore the words 'topos' and 'internal logic' and imagine that the statements and proofs are about PERs in the ordinary sense; we then obtain results similar in flavour to those presented in [1], though of course the underlying model is different and so the same techniques cannot be applied.

But note that results proved using the internal logic of a topos yield, when viewed externally, apparently stronger 'indexed' or 'uniform' results, which are often exactly what we need to model theories of dependent types or the calculus of constructions. So in this case, the use of topos theory is not just conceptually attractive; it's saving us work as well.

Even more is true. Completeness in an appropriate sense ensures reflectivity; and this has good consequences. For example, the categories we will define admit a *convex powerdomain* construction, which exists for quite a trivial reason. This is not something we can deduce immediately if we insist on taking an external point of view, speaking only of closure under intersection, and so on....

This does not change the fact that this paper relies crucially on topos theory and on the theory of realizability toposes. It's unfortunate that no really accessible account exists. This situation should soon be rectified: a survey paper by Nordemann and Knijnenburg and, separately, an exposition by the present author are in preparation.

Summary of results Section 2 contains some background on realizability toposes and modest sets, including a couple of general results that do not seem to have appeared elsewhere. There is also a sketch of Rosolini's theory of *dominances* (see [30]) which we will later use to isolate a nice class of partial maps in the model. As most of the material in this section is not (essentially) original, few proofs are included.

The main results appear in section 3. The goal is to unify domain theory and the λ-calculus: not by using domains to model the λ-calculus, but by showing that the λ-calculus gives rise to categories of 'domains'. The whole approach relies on the basic fact that application is continuous with respect to the tree topology; the aim is to lift this result to a suitable class of PERs.

We work inside a certain realizability topos based on a particular (total) combinatory algebra: the closed term model B of the λ-calculus in which terms having the same Böhm tree are identified. This turns out to be just the right identification to make to get the continuity properties we need. Our reference for facts about this model, and the λ-calculus in general, is the book [3].

In contrast to, say, [20] or [11], this paper has quite a concrete flavour: we are interested in PERs themselves, and in the relationship between the type structure and the untyped computations. The idea is that computation really goes on at an untyped level—the type structure is just semantic scaffolding that falls away at run-time—and the theory should reflect this.

First we isolate a suitable notion of 'computable partial map' applicable to any object in the topos. (We need to do this so that we can model non-terminating programs.) It's enough to find an appropriate dominance. The computable partial map classifier of an object will turn out to be the correct lifted domain construction.

The next step is to define a category of predomains. We say what it means for a PER (a modest set) to be **complete**—that is, to have 'sups of chains'—and show that *all* maps between complete PERs automatically preserve these sups. This is *not* achieved by defining a partial order on the PER, but more directly: the method is based on the idea of continuity in the sense of Böhm trees. Anyway, these PERs will be our predomains.

We say that a PER is **canonically focal**—has a 'bottom element'—if the unsolvable term is in its domain. It turns out that a complete, canonically focal PER has fixed points of endofunctions—and these may be calculated using the fixed point combinator y, as we might hope. Such a PER can thus be regarded as a domain.

An odd feature of our predomains is that there seems to be no natural, well-behaved partial order that we can impose on them. There *are* natural candidates for an intrinsic partial order relation, as in other models such as the effective topos, but we'll see in section 3.4 that previous techniques for proving things about the intrinsic order necessarily break down here. Moreover there are good reasons for believing that no 'well-behaved' preorder can be defined internally, though it's impossible to give a proof of a vague assertion like this. In any case it is interesting that the partial order structure is *not needed,* and one can sensibly work with a category of domains which is not cpo-enriched.

The categories of predomains is, in an appropriate sense (see [14], [29]), *small* and *complete*. As mentioned before, this means that it's possible to construct polymorphic types; they provide models for the theory of constructions ([7], [13]; we can only model weak, not strong, sums). It's also possible to solve covariant domain equations (and hence to model recursively defined algebraic data types); and to construct convex powerdomains as free semilattices, in the way that was sketched in [20].

In section 4, we sketch an interpretation of (call-by-name) PCF in our category of domains. This is related to a certain translation of PCF into the untyped λ-calculus, and the fact that the interpretation is computationally adequate corresponds to a previously unknown fact about this translation.

In the conference version of this paper, a notion of 'meet-closure' for PERs was also defined; it was shown that the category of meet-closed PERs was reflective, and that all maps between meet-closed PERs were 'stable'. Although these results are suggestive, they have been omitted here—this is because I know of no concrete applications.

There are other results about the predomains described here which had to be omitted because of lack of space (and time). In particular, they provide fairly simple models of systems with subtyping and recursive types, such as Quest. There are also quite a few open questions, especially regarding the relationships with operational semantics.

Note. The theorems we'll be proving are nearly all statements in the internal language. However, not all the proofs can be carried out entirely in the internal language; we'll need to use particular properties of the model, together with the characterization of validity in the internal language of a realizability topos as explained in [9]. Unfortunately, there is only room for a very brief exposition in section 2. This is not meant to be a substitute for the original papers.

2 Realizability toposes and modest sets

2.1 Realizability toposes

I have to assume some familiarity with topos theory. But for those who know nothing about it: a topos $\mathcal{E}$ is a category with some additional structure that lets us regard it, very roughly speaking, as a model of constructive set theory. If we are working inside a fixed topos $\mathcal{E}$, we often abuse language by calling its objects 'sets'.

Any topos $\mathcal{E}$ has an *internal logic*: (almost) any set-theoretic or mathematical statement ϕ can be interpreted in $\mathcal{E}$, and it makes sense to say that ϕ is or is not 'true'—i.e. valid—in $\mathcal{E}$. And in general, any statement that is constructively provable is valid in any topos; for example,

$$\forall n \in \mathbb{N}.\, \exists p \in \mathbb{N}.\, \text{“}p \text{ is prime”} \,\wedge\, p > n.$$

However, a classically provable statement need not be valid in an arbitrary topos; for example, $\neg\neg\phi \to \phi$ and $\phi \vee \neg\phi$ will not in general be valid.

Definition 2.1.1 *A* ***partial combinatory algebra*** A *is a set equipped with a partial binary operation* $\mathsf{A} \times \mathsf{A} \rightharpoonup \mathsf{A}$*, 'application', and with designated elements* $\mathsf{s}, \mathsf{k} \in \mathsf{A}$ *satisfying the usual axioms.*

For example, any model of the λ-calculus is a partial (in fact, a total) combinatory algebra; another example would be the partial recursive functions.

We'll sometimes talk about a combinator or a closed λ-term as if it were an element of A; what we really mean is the element of A that it denotes. For example, when we write i or $\lambda x.x$ we really mean $\mathsf{skk} \in \mathsf{A}$.

The particular toposes we are concerned with will be constructed using partial combinatory algebras. A statement of set theory is valid in such a topos exactly when there is an element of the algebra (an operation) that computes evidence for its truth. The logic we must use is constructive just because we have this computational interpretation of truth.

Thought of as models of constructive set theory, these toposes extend the realizability interpretation of constructive logic. For example, to say that

$$\forall n \in \mathbb{N}.\, \exists p \in \mathbb{N}.\, \text{“}p \text{ is prime”} \wedge p > n$$

is valid amounts to saying that there exists an element $a \in \mathsf{A}$ such that for any Church numeral c_n, $a\mathsf{c}_n$ is a Church numeral c_p where p is a prime $> n$. See 2.2 for details.

The construction of the realizability topos $\mathcal{E}$ associated to a partial combinatory algebra A is due to Hyland: see [9] Powell had the same idea a little earlier, but his work has never appeared in print; in any case, Hyland's categorical framework will be more useful for the purposes of this paper.

Note. Different partial combinatory algebras do give rise to inequivalent realizability toposes, even when they can represent the same numerical functions—so it's inaccurate to talk about *the* 'realizability universe'. Unfortunately, the relationship between a partial combinatory algebra and the internal logic of its realizability topos is, except in special cases, still poorly understood.

Assemblies Rather than discussing the construction of $\mathcal{E}$ (which is sketched in 2.2) it's helpful to describe a subcategory of $\mathcal{E}$, the category of *A-valued assemblies*, that's more familiar and easier to think about. Although the proofs in the paper rely on the internal language of $\mathcal{E}$, they may be understood more concretely, in terms of PERs on A and A-valued assemblies.

Definition 2.1.2 *Let* A *be a combinatory algebra. An* ***A-valued assembly*** *(or just an* ***assembly*** *if* A *is understood) consists of a set* X *together with, for each* $x \in X$*, a nonempty subset* $|x \in X| \subseteq \mathsf{A}$*. If* X *and* Y *are two assemblies, a map* $X \to Y$ *is defined to be a function* $f \colon X \to Y$ *between the underlying sets, for which there exists an element* $a \in \mathsf{A}$ *such that for all* $x \in X$*,*

$$p \in |x \in X| \longrightarrow ap \in |f(x) \in Y|$$

(we say that a ***tracks*** f*). The category of (*A*-valued) assemblies is called* $\mathcal{S}ep$*.*

We think of $|x \in X|$ as a set of 'codes for x'; as x varies, these subsets need not necessarily be disjoint. (In the case $\mathsf{A} = \mathbb{N}$, the indices for partial recursive functions, A-valued assemblies are usually called **ω-sets**; they have been widely used in the semantics of various typed λ-calculi.) Assemblies appear already in [9], where they are called 'canonically separated objects', and the idea dates back to Ershov; their relevance in computer science was recognised by Longo and Moggi. More information may be found in [18] and [6].

$\mathcal{S}ep$ (the terminology is explained by 2.1.7) has finite limits and is cartesian closed. It also contains the ordinary category of sets—there is a full embedding $\mathcal{S}ets \hookrightarrow \mathcal{S}ep$: given a set X, for all $x \in X$ put

$$|x \in X| = \mathsf{A}$$

This has a left adjoint, namely the functor that takes an assembly to its underlying set. On the other hand, there are plenty of objects which are not in the image of this embedding. For example,

Definition 2.1.3 *A itself gives rise to an assembly $\mathbb{A}$, with underlying set A and*

$$|a \in \mathbb{A}| = \{a\}$$

for all $a \in \mathsf{A}$.

$\mathcal{S}ep$ is not a topos. The authors of [6] show how to construct the realizability topos $\mathcal{E}$ out of $\mathcal{S}ep$ in a fairly simple way, by adjoining 'good' quotients of equivalence relations. However, for our calculations in 3.3 it's more convenient to work with Hyland's original description, see 2.2.

Some remarks on the logic Any topos contains an object Ω of truth-values, endowed with an entailment relation $\leq$, and with a distinguished element $\top$ ('true'). For example, in $\mathcal{S}ets$ (the ordinary category of sets), $\Omega = \{\top, \bot\}$ where $\bot \leq \top$.

In the realizability topos $\mathcal{E}$ arising from a partial combinatory algebra A, Ω looks roughly as follows: the truth-values are subsets of A, and if $p, q \subseteq \mathsf{A}$ then $p \leq q$ exactly when there is an element $a \in \mathsf{A}$ such that $ab \in q$ for all $b \in p$. This is not a precise description of Ω; in fact, as Ω is not an assembly, we aren't in a position to describe it precisely yet. See 2.2.)

If X is an object of $\mathcal{E}$, we can regard a mono $X' \rightarrowtail X$ as giving us a 'subset' of X. For each such mono there is a unique map $\phi: X \to \Omega$, its characteristic function, such that the following square is a pullback:

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

The map $\phi: X \to \Omega$ can be though of as a predicate on X, whose extension is X'. In other words, we can think of X' as being

$$\{x | \phi(x)\} \rightarrowtail X$$

(and in fact, we often define subsets of X by specifying predicates on X). A predicate $\phi: X \to \Omega$ is 'true' or valid if $X' = X$; that is, if ϕ is the map

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

Definition 2.1.4 *Let $\phi: X \to \Omega$ be a predicate on X. Then ϕ is* ***$\neg\neg$-closed*** *if*

$$\forall x \in X.\, \neg\neg\phi(x) \to \phi(x)$$

is valid.

This definition is not vacuous: because the logic of $\mathcal{E}$ is constructive, if ϕ is a predicate on X then ϕ and $\neg\neg\phi$ are not in general the same. If X is an A-valued assembly, thought of as an object of $\mathcal{E}$, the $\neg\neg$-closed predicates on X are in bijection with the (ordinary) subsets of the underlying set of X. More general predicates on X correspond to subsets whose elements are endowed with extra intensional information.

The following examples may help.

Definition 2.1.5 *Any $C \subseteq \mathsf{A}$ is the underlying set of an assembly $\bar{C}$ defined in a natural way: if $a \in C$, just put*

$$|a \in \bar{C}| = \{a\}$$

The identity element $\mathbf{i} = \mathbf{skk} \in \mathsf{A}$ tracks a mono $\bar{C} \rightarrowtail \mathbb{A}$ in $\mathcal{S}ep$ (and so in $\mathcal{E}$, since $\mathcal{S}ep$ is a full subcategory of $\mathcal{E}$) and this mono is classified by a $\neg\neg$-closed predicate $\phi\colon \mathbb{A} \to \Omega$.

For example, let A be N, the set of indices for partial recursive functions; the corresponding realizability topos is called $\mathcal{E}\!f\!f$, the effective topos. As in 2.1.3, N is the underlying set of an assembly $\mathbb{N}$, where we simply put

$$|n \in \mathbb{N}| = \{n\}$$

And as above, any (ordinary) subset $C \subseteq \mathsf{N}$ gives us a $\neg\neg$-closed subobject $\bar{C} \rightarrowtail \mathbb{N}$ in $\mathcal{E}$, classified by a $\neg\neg$-closed predicate ϕ on $\mathbb{N} \in \mathcal{E}$.

However, $\phi \vee \neg\phi$ is not in general a $\neg\neg$-closed predicate on $\mathbb{N}$. It corresponds to a mono $\bar{C} \vee \neg\bar{C} \rightarrowtail \mathbb{N}$, where $\bar{C} \vee \neg\bar{C}$ denotes the assembly with underlying set N where we put

$$|n \in \bar{C} \vee \neg\bar{C}| = \begin{cases} \{\langle n, 0\rangle\} & n \notin C \\ \{\langle n, 1\rangle\} & n \in C \end{cases}$$

$\bar{C} \vee \neg\bar{C} \rightarrowtail \mathbb{N}$ is an isomorphism only when

$$\forall n \in \mathbb{N}.\, n \in \bar{C} \vee n \notin \bar{C}$$

is valid; i.e. if C is a recursive subset of N.

There is one particular example of a $\neg\neg$-closed predicate that is worth mentioning. The diagonal inclusion $\mathsf{A} \rightarrowtail \mathsf{A} \times \mathsf{A}$ gives us a mono $\mathbb{A} \rightarrowtail \mathbb{A} \times \mathbb{A}$ whose classifying map is the equality predicate $\mathbb{A} \times \mathbb{A} \to \Omega$ taking $\langle a, b\rangle \mapsto (a = b)$. So the equality predicate on $\mathbb{A}$ is $\neg\neg$-closed:

$$\neg\neg(a = b) \;\to\; (a = b).$$

Definition 2.1.6 *An object $X \in \mathcal{E}$ is **$\neg\neg$-separated** if*

$$\forall x, x' \in X.\, \neg\neg(x = x') \to (x = x')$$

is valid in $\mathcal{E}$.

Proposition 2.1.7 *(See [9, section 6].) The category of $\neg\neg$-separated objects in $\mathcal{E}$ is equivalent to the category of A-valued assemblies.*

It follows immediately that $\mathcal{S}ep$ is reflective in $\mathcal{E}$, that it is a locally cartesian closed category, and so on. $\mathcal{S}ets$ is also a full reflective subcategory of $\mathcal{E}$, via the embedding $\mathcal{S}ets \hookrightarrow \mathcal{S}ep$; it can be identified with the category of so-called $\neg\neg$-sheaves.

2.2 Realizability and the internal language

For later use, we recall some facts about the logic of $\mathcal{E}$; for brevity, we'll oversimplify matters a little (the full details are in [9] and [12]). This material is only needed in one proof—see 3.3—and can safely be skipped.

Formulas ϕ are realized by elements of A; in other words, the 'truth-values' in the logic are subsets of A, i.e. elements of the powerset $\mathbf{P}\mathsf{A}$. We can use $\mathbf{P}\mathsf{A}$ to interpret logical connectives by making it into a Heyting prealgebra in an appropriate way: for example, we can let $\top = \mathsf{A}$ and $\bot = \varnothing$; and if $p, q \in \mathbf{P}\mathsf{A}$ then we define

$$\begin{array}{rcl} p \wedge q & = & \{\langle a, b\rangle \mid a \in p \text{ and } b \in q\} \\ p \rightarrow q & = & \{a \mid \text{for all } b \in p,\ ab \in q\} \end{array}$$

(Here $\langle a, b\rangle$ denotes $\mathsf{p}ab$ where p is some element of A that acts as a pairing combinator.)

Objects of $\mathcal{E}$ are pairs $(X, =)$ where X is a set and $=$ is a '$\mathbf{P}\mathsf{A}$-valued equality relation' on X; i.e. for each pair of elements $x, x' \in X$ we have a truth-value $|x = x'| \in \mathbf{P}\mathsf{A}$ [such that certain conditions are satisfied...] For example, any assembly X gives rise to an object $(X, =)$ where

$$|x = x'| = \begin{cases} |x \in X| & x = x' \\ \varnothing & x \neq x' \end{cases}$$

and Ω is the object $(\mathbf{P}\mathsf{A}, \leftrightarrow)$ where if $p, q \in \mathbf{P}\mathsf{A}$,

$$|p \leftrightarrow q| = (p \rightarrow q) \wedge (q \rightarrow p).$$

If $(X, =) \in \mathcal{E}$, we abbreviate $|x = x|$ by $|x \in X|$; this notation is consistent with the way objects of $\mathcal{E}$ can be constructed out of assemblies.

Predicates ϕ on $(X, =)$ correspond to maps $X \rightarrow \mathbf{P}\mathsf{A}$ [satisfying certain conditions...]; that is, for each $x \in X$ we are given a set $|\phi(x)| \in \mathbf{P}\mathsf{A}$, the set of **realizers** for $\phi(x)$. So, for example,

$$\begin{array}{rcl} |(\phi \wedge \psi)(x)| & = & |\phi(x)| \wedge |\psi(x)| \\ |(\phi \rightarrow \psi)(x)| & = & |\phi(x)| \rightarrow |\psi(x)| \end{array}$$

and

$$\begin{array}{rcl} |\forall x.\phi(x)| & = & \bigcap_{x \in X} |x \in X| \rightarrow |\phi(x)| \\ |\exists x.\phi(x)| & = & \bigcup_{x \in X} |x \in X| \wedge |\phi(x)| \end{array}$$

We say ϕ is **valid** if $|\forall x.\phi(x)| \neq \varnothing$; this is equivalent to saying that ϕ is valid in the internal language of $\mathcal{E}$.

2.3 Modest sets and completeness

Definition 2.3.1 *An assembly is a* ***modest set*** *if two different elements never share a common code; that is, if for all $x, x' \in X$,*

$$x \neq x' \longrightarrow |x \in X| \cap |x' \in X| = \varnothing.$$

Let $\mathcal{M}od$ be the full subcategory of $\mathcal{S}ep$ whose objects are the modest sets.

$\mathcal{M}od$ contains, for example, the object $\mathbb{A}$.

Definition 2.3.2 *An (internal)* PER *on* $\mathbb{A}$ *in* $\mathcal{E}$ *is a relation* $R \rightarrowtail \mathbb{A} \times \mathbb{A}$*, or equivalently* $R \in P(\mathbb{A} \times \mathbb{A})$*, such that*

$$\forall a, a' \in \mathbb{A}.\, aRa' \rightarrow a'Ra$$
$$\forall a, a', a'' \in \mathbb{A}.\, aRa' \wedge a'Ra'' \rightarrow aRa''$$
$$\forall a, a' \in \mathbb{A}.\, \neg\neg(aRa') \rightarrow aRa'$$

are valid. (Here PX *is the internal powerset* Ω^X*.)*

Up to isomorphism, the modest sets are exactly the quotients of $\mathbb{A}$ in $\mathcal{E}$ under internal PERs on $\mathbb{A}$.

The category of modest sets is essentially the same as the category Per of ordinary PERs on A: for ordinary PERs $R \subseteq \mathsf{A} \times \mathsf{A}$ are in bijection with internal PERs on $\mathbb{A}$ (cf. 2.1.5).

In terms of assemblies: notice that any ordinary PER R gives rise to an assembly, where we have

- underlying set $\mathsf{A}/R = \{R\text{-equivalence classes}\}$;
- if $u \in \mathsf{A}/R$, $|u \in \mathsf{A}/R| = u \subseteq \mathsf{A}$.

That is, the codes for an equivalence class are the elements of the equivalence class. Then we get another full embedding $\iota\colon \mathsf{Per} \hookrightarrow \mathcal{S}ep$ that preserves finite limits and the cartesian closed structure; and the image of ι is equivalent to $\mathcal{M}od$.

Modest sets in $\mathcal{E}$ Because $\mathcal{S}ep$ is a full subcategory of $\mathcal{E}$, a modest set can be regarded as a 'set' in $\mathcal{E}$. There is then an internal characterisation of maps between modest sets which can be thought of as a generalisation of Church's thesis, cf. [9, proposition 10.4].

First note that any $a \in \mathsf{A}$ gives rise to a map

$$[a]\colon \mathbb{A} \rightarrow \mathbb{A}$$

sending $b \mapsto ab$. One can check that

$$\forall f \in \mathbb{A}^{\mathbb{A}}. \exists a \in \mathbb{A}.\, f = [a]$$

is valid in $\mathcal{E}$, i.e. that every map $\mathbb{A} \rightarrow \mathbb{A}$ arises in this way; this can be seen as a partial analogue of Church's thesis. (However, Church's thesis will *not* in general be valid in $\mathcal{E}$.)

Let X be a modest set and let

$$A = \{a \in \mathsf{A} \,|\, a \in |x \in X| \text{ for some } x \in X\}$$

Then we have a diagram

$$X \overset{q_X}{\twoheadleftarrow} \bar{A} \rightarrowtail \mathbb{A}$$

where $\bar{A}$ is the $\neg\neg$-closed subobject of $\mathbb{A}$ determined by A as in 2.1.5. Note that we then have $X \cong \mathbb{A}/R$ where R is a $\neg\neg$-closed PER on $\mathbb{A}$ with $\operatorname{dom} R = A$.

Let $X = \mathbb{A}/R, Y = \mathbb{A}/S$ be modest sets, which are quotients of $\neg\neg$-closed subobjects $\bar{A}, \bar{B} \rightarrowtail \mathbb{A}$ as just described. Observe that $a \in \mathbb{A}$ tracks a map $X \rightarrow Y$ exactly when

$$\forall c, c' \in \bar{A}.\, cRc' \rightarrow acSac'$$

In other words, the formula

$$\forall a \in \mathbb{A}.\, ((\forall c, c' \in \bar{A}.\, cRc' \rightarrow acSac') \longrightarrow (\exists! f \in Y^X. \forall c \in \bar{A}.\, f(q_X(c)) = q_Y(ac)))$$

is valid. In fact, *all* maps arise in this way:

Proposition 2.3.3 *In the situation just described,*

$$\forall f \in Y^X. \exists a \in \mathbb{A}. \forall c \in \bar{A}.\ f(q_X(c)) = q_Y(ac)$$

is valid in $\mathcal{E}$.

This is the *internal* sense in which all maps between modest sets are 'tracked' by elements of $\mathbb{A}$. We will use this fact frequently.

There is actually a more uniform way of stating this, which quantifies over PERs as well as over maps:

$$\begin{aligned}&\forall R, S \in P(\mathbb{A} \times \mathbb{A}).\ \text{“}R, S \text{ are PERs on } \mathbb{A}\text{”} \rightarrow \\ &\qquad \forall f \in (\mathbb{A}/S)^{(\mathbb{A}/R)}. \exists a \in \mathbb{A}. \forall c \in A.\ f(q_X(c)) = q_Y(ac)\end{aligned}$$

is valid in $\mathcal{E}$. Here "R is a PER on $\mathbb{A}$" is an abbreviation for the conjunction of the formulas in definition 2.3.2.

The internal category of modest sets The important point is that there is an internal category $\mathbb{M}\mathbf{od}$ in $\mathcal{E}$—the category of $\neg\neg$-closed partial equivalence relations on $\mathbb{A}$ and maps between them—whose 'externalization' (in a certain sense, see [14]) is $\mathcal{M}od$. The object-of-objects of $\mathbb{M}\mathbf{od}$ is the following subobject of $P(\mathbb{A} \times \mathbb{A})$:

$$\mathbb{M}\mathbf{od}_0 = \{R \in P(\mathbb{A} \times \mathbb{A}) |\ \text{“}R \text{ is a PER on } \mathbb{A}\text{”}\}$$

One explanation of why the PER model of polymorphism works is that $\mathbb{M}\mathbf{od}$ is, in a suitable sense (see [14] and [30]), *complete*; in fact, it is closed under all limits in $\mathcal{E}$. Hence, for example, polymorphic types $\Pi X.T(X)$ can be taken to be $\mathbb{M}\mathbf{od}_0$-indexed products $\Pi_{X \in \mathbb{M}od} T(X)$. This is the so-called Moggi-Hyland 'set-theoretic' interpretation of polymorphism. (Other interpretations based on modest sets do exist, some of them better behaved.)

Incidentally, Moggi made the crucial observation that the PER whose quotient is the Moggi-Hyland polymorphic type $\Pi_{X \in \mathbb{M}od} T(X)$ can actually be calculated as an *intersection*

$$\bigcap_{\text{“}R \text{ a PER on } \mathbb{A}\text{”}} T(R)$$

(You can think of this intersection as being computed internally, as here, or externally over all ordinary PERs on A; the answer you get is the same.) This is not necessarily true for complete full subcategories of $\mathbb{M}\mathbf{od}$, such as various categories of synthetic domains. However, it *will* hold for the category of domains studied in this paper.

Note. Actually, we need the language of indexed categories or fibrations to formulate all this properly; everything ought to be regarded as a fibration: see [14]. The theory of fibrations won't be discussed here; what this approach amounts to, though, is considering families $\{X_i\}_{i \in I}$ of modest sets 'indexed' by arbitrary objects $I \in \mathcal{E}$; such families are given as objects $\binom{X}{\downarrow \atop I}$ in the slice topos $\mathcal{E}/I$. By 'completeness' we mean, for example, closure under 'products' indexed by objects $I \in \mathcal{E}$. See also [29].

In fact, it is only those families indexed by assemblies $I \in \mathcal{S}ep$ that are well-behaved, and $\mathcal{M}od$ is really only complete (in the sense appropriate to fibrations) indexed over $\mathcal{S}ep$; however, this is enough to deduce all the results we need. For example, $\mathbb{M}\mathbf{od}$ is an internal category in $\mathcal{S}ep$, not just in $\mathcal{E}ff$, and in particular $\mathbb{M}\mathbf{od}_0 \in \mathcal{S}ep$; so the Moggi-Hyland interpretation does carry through.

2.4 Partial maps and dominances

If we restrict ourselves to the ordinary category of modest sets, then all maps are computable; that is, all *total* functions. But (as usual in computer science) we will also be interested in partial functions; and because the notion of subobject in a realizability topos is very flexible, there are many partial maps which we don't want to regard as computable. For example, in the case $\mathsf{A} = \mathsf{N}$, if $C \subseteq \mathsf{N}$ is an arbitrary subset then there is a partial map $\mathbb{N} \rightharpoonup 2$ (with domain $\bar{C} \vee \neg\bar{C}$) that is the characteristic function of $\bar{C} \rightarrowtail \mathbb{N}$. It makes use of the 'intensional information' in the description of $\bar{C} \vee \neg\bar{C}$ and is not the kind of partial map that arises in programming language semantics.

We want to be able to isolate a smaller class of computable partial maps. Building on previous theories of partial maps in general categories (due to Heller and others) and influenced particularly by ideas of Mulry and Scott, Rosolini developed a convenient framework for defining classes of partial maps in a topos. The idea is to single out an appropriate subset $\Sigma \rightarrowtail \Omega$ of truth-values—which may be regarded as the 'computable' truth-values. A subobject $X' \rightarrowtail X$ whose characteristic function takes values in Σ will be called a **Σ-subset** (we then write $X' \subset_\Sigma X$); it will turn out that—if we choose the right Σ—the partial maps whose domains are Σ-subsets are exactly the ones which can be regarded as computable. We'll call them **Σ-partial maps**.

Definition 2.4.1 *Let $\Sigma \rightarrowtail \Omega$ be a subset of the object of truth-values. Then Σ is a **dominance** if*

$$\top \in \Sigma\,;$$
$$p \in \Sigma \,\wedge\, (p = \top \rightarrow q \in \Sigma) \,\rightarrow\, (p \wedge q) \in \Sigma.$$

are valid

These conditions ensure that the Σ-partial maps form a category: they contain identity maps and are closed under composition.

It may be more enlightening to cast these conditions in an 'external' form. Stated this way, they say that

- $X \subset_\Sigma X$ for any object X; and
- if $X \subset_\Sigma Y$ and $Y \subset_\Sigma Z$ then $X \subset_\Sigma Z$.

These conditions have to be interpreted in a suitable 'indexed' or fibred sense, however; that is, we have to talk about (internal) families of objects, and families of Σ-subsets. See below.

The most well-known example of a dominance is the *r.e. subobject classifier* (see [30]) in the effective topos $\mathcal{E}\!f\!f$. Σ-subsets in this case are a generalization of r.e. subsets—we can talk about a Σ-subset of any object of $\mathcal{E}\!f\!f$, not just of $\mathbb{N}$—and Σ-partial maps are generalized partial recursive functions.

Once we've fixed a dominance Σ, we have a notion of partial map classifier:

Definition 2.4.2 *Given an object $X \in \mathcal{E}$, we say that another object $X_\perp$ equipped with a mono $X \rightarrowtail X_\perp$ is a Σ-**partial map classifier** for X if, for any object Y, there is a natural bijection between Σ-partial maps $Y \rightharpoonup X$ and (total) maps $Y \rightarrow X_\perp$. More precisely, given a partial map*

$m: Y' \rightarrowtail Y$, $f: Y' \to X$, there is a unique map $\hat{f}: Y \to X_\perp$ such that we have a pullback

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

*The object $X_\perp$ is necessarily unique up to isomorphism; it is also called the **lift** of X.*

Note that we then have partial function spaces: the Σ-partial function space $[Y \rightharpoonup X]$ is just the ordinary exponential ${X_\perp}^Y$.

There are several general methods for constructing the Σ-partial map classifier $X_\perp$; rather than give a categorical construction here, we will give an explicit description in 3.2 once we've defined the particular dominance Σ we need.

Relative dominances The preceding remarks apply when we want to talk about computable partial maps between any two objects of the topos. However, it's sometimes the case that we're only interested in a particular subcategory $\mathcal{C}$ of a topos: for example, $\mathcal{S}ep$ or $\mathcal{M}od$. In this case we would only require that Σ satisfy the (external) conditions for being a dominance in the case where all the objects lie in $\mathcal{C}$; and similarly, we would require that for $X \in \mathcal{C}$ there was an object $X_\perp \in \mathcal{C}$ satisfying an appropriate universal property for Σ-partial maps $Y \rightharpoonup X$ with $Y \in \mathcal{C}$. We could say that we had a dominance, and partial map classifiers, *relative to* $\mathcal{C}$.

Again, these statements have to be read in an appropriate 'indexed' sense; let's spell this out in the case when the category we're interested in is $\mathcal{S}ep$. We assume that Σ is an object of $\mathcal{S}ep$; note that for each $I \in \mathcal{S}ep$ we have the object $\binom{\Sigma\times I}{\downarrow \atop I} \in \mathcal{S}ep/I$, which classifies I-indexed families of Σ-subsets.

Definition 2.4.3 *Σ is a **dominance relative to $\mathcal{S}ep$** if*

- *$\binom{X}{\downarrow \atop I} \subset_\Sigma \binom{X}{\downarrow \atop I}$ for any $\binom{X}{\downarrow \atop I} \in \mathcal{S}ep/I$; and*
- *if $\binom{X}{\downarrow \atop I} \subset_\Sigma \binom{Y}{\downarrow \atop I}$ and $\binom{Y}{\downarrow \atop I} \subset_\Sigma \binom{Z}{\downarrow \atop I}$ then $\binom{X}{\downarrow \atop I} \subset_\Sigma \binom{Z}{\downarrow \atop I}$.*

In fact, usually verifying these conditions for general I is just as easy as verifying them in the case $I = 1$, so long as the arguments are sufficiently 'uniform'.

2.5 Some subobjects of $\mathbb{A}$ that may be dominances

In the present paper (as in [21], though not [20]) Σ will be a $\neg\neg$-closed subobject of $\mathbb{A}$ arising from a subset

$$\{a, b\} \subseteq \mathsf{A}$$

as in 2.1.5. That is, it will be a modest set with underlying set $\{\perp, \top\}$ and

$$|\perp \in \Sigma| = \{a\}, \qquad |\top \in \Sigma| = \{b\}$$

Σ can also be defined in the internal logic by a predicate on $\mathbb{A}$; we need to be careful to write down one that is $\neg\neg$-closed, as required. We put

$$\Sigma = \{c \in \mathbb{A} | \neg(c = a) \to (c = b)\}$$

(Not all such modest sets will necessarily be dominances!)

Note that we have maps $\bot, \top: 1 \to \Sigma$, and

$$\forall x \in \Sigma. \neg(x = \bot) \to (x = \top)$$

is valid, cf. [9]. We must verify that Σ can actually be regarded as a subobject of Ω, i.e. as a subset of truth-values, in such a way that $\bot, \top$ can be regarded as the usual truth-values 'false' and 'true' in Ω.

Proposition 2.5.1 *Σ as defined above is isomorphic to the subset*

$$\Sigma_1 \; = \; \{p \in \Omega | \, \exists c \in \mathbb{A}. (\neg p \leftrightarrow (c = a)) \wedge (p \leftrightarrow (c = b))\}$$

of Ω.

Proof. First note that if

$$\exists c \in \mathbb{A}. (\neg p \leftrightarrow (c = a)) \wedge (p \leftrightarrow (c = b))\}$$

then, as one of the referees pointed out, such a c is unique. For (arguing informally, in the internal language) suppose we have $d \in \mathbb{A}$ such that

$$(\neg p \leftrightarrow (d = a)) \wedge (p \leftrightarrow (d = b))$$

If $\neg p$ then $c = a = d$, and if p then $c = b = d$, so

$$(p \vee \neg p) \to c = d$$

But equality on $\mathbb{B}$ is $\neg\neg$-closed, so

$$\top = \neg\neg(p \vee \neg p) \to c = d$$

So we may assume that

$$\Sigma_1 \; = \; \{p | \, \exists! c \in \mathbb{A}. (\neg p \leftrightarrow (c = a)) \wedge (p \leftrightarrow (c = b))\}$$

There is a map $\Sigma_1 \to \Sigma$ taking $p \mapsto c$. We can also define a map $\Sigma \to \Omega$ taking $c \mapsto |c = b|$; this actually maps into Σ_1 (we must check that $\neg(c = b) \leftrightarrow (c = a)$ and $(c = b) \leftrightarrow (c = b)$, which is easy). These two maps are mutually inverse. ∎

Now, let's see what it means for a subobject of a modest set—which must necessarily be a modest set—to be a Σ-subset. Suppose we have a mono $m: X \rightarrowtail Y$ where X, Y are modest. This defines a Σ-subset if there is a map $\phi: Y \to \Sigma$ such that we have a pullback square

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

We say that m **displays** a Σ-subset. This is equivalent to the validity of

$$\exists \chi \in \Sigma^Y . \forall y \in Y. (\chi(y) = \top) \leftrightarrow (\exists x \in X. m(x) = y)$$

(note that such a map χ is provably unique, so this formula does imply the external existence of a suitable map $\chi: Y \to \Sigma$).

Letting

$$X \overset{q_X}{\twoheadleftarrow} \bar{A} \rightarrowtail \mathbb{A}, \qquad Y \overset{q_Y}{\twoheadleftarrow} \bar{B} \rightarrowtail \mathbb{A}$$

and supposing m is tracked by $b \in \mathbb{A}$, we can unwind this further.

Proposition 2.5.2 *$X \subset_\Sigma Y$ (via m) exactly when*

$$\exists d \in \mathbb{A}. (\text{“}d \text{ tracks a map } Y \to \Sigma\text{”}) \wedge (\text{“}d \text{ tracks the classifying map of } m\text{”})$$

is valid, where “d tracks a map $Y \to \Sigma$” means

$$(\forall c \in \bar{B}. \neg(dc = a) \to (dc = b)) \wedge (\forall c, c' \in \bar{B}. q_Y(c) = q_Y(c') \to dc = dc')$$

and “d tracks the classifying map of m” means

$$\forall c \in \bar{B}. (dc = b) \leftrightarrow (\exists x \in X. m(x) = q_Y(c))$$

So if we're given X, Y as quotients of $\mathbb{A}$ by $\neg\neg$-closed PERs, showing that $X \subset_\Sigma Y$ amounts to showing that there exists $d \in \mathbb{A}$ satisfying certain conditions.

2.6 A criterion for reflectivity

In [20], the category of complete Σ-spaces was shown to be complete; one could deduce from this that it was reflective—see [22] for more of the technical details. In this paper we will be working the other way: we'll show that certain categories of PERs (i.e. subcategories of $\mathbb{M}\mathbf{od}$) are reflective, and deduce that they are complete.

The following result will be a useful tool. It does not seem to have been observed before, though it is quite simple. Note that all the statements are meant to be interpreted in the internal language.

Proposition 2.6.1 *Let $\mathbb{C}_0$ be an (internal) collection of PERs such that*

- *if $(R_i)_{i \in I}$ is any internal family of PERs in $\mathbb{C}_0$, then their intersection $\bigcap_{i \in I} R_i$ is also in $\mathbb{C}_0$;*
- *if $a \in \mathbb{A}$ and $R \in \mathbb{C}_0$ then $a^{-1}R \in \mathbb{C}_0$, where $a^{-1}R$ is defined by*

$$p(a^{-1}R)q \qquad \longleftrightarrow \qquad (ap)R(aq);$$

- *the full subcategory $\mathbb{C}$ of $\mathbb{M}\boldsymbol{od}$ with object-of-objects $\mathbb{C}_0$ is closed under the formation of equalizers in $\mathbb{M}\boldsymbol{od}$; that is, if $R, S \in \mathbb{C}_0$ and $a, b \in \mathbb{A}$ track maps $\mathbb{A}/R \rightrightarrows \mathbb{A}/S$ then the PER E is in $\mathbb{C}_0$, where*

$$pEq \qquad \longleftrightarrow \qquad (pRq \wedge (ap)S(bq)).$$

Then there is a reflection functor $\mathbb{M}\boldsymbol{od} \to \mathbb{C}$.

Proof. (Sketch.) Let R be any PER in $\mathbb{M}\mathbf{od}_0$; we'll construct a PER $\tilde{R}$ that's its reflection into $\mathbb{C}$. In fact, let

$$\tilde{R} = \bigcap_{S \in \mathbb{C}_0, R \subseteq S} S$$

or, more precisely

$$\tilde{R} = \{\langle x, y\rangle \mid \forall S \in C_0.\, R \subseteq S \to \langle x, y\rangle \in S\}$$

where $R \subseteq S$ means

$$\forall x, y \in \mathbb{A}.\, \langle x, y\rangle \in R \to \langle x, y\rangle \in S$$

Note that $\tilde{R}$ is a $\neg\neg$-closed partial equivalence relation (an easy exercise in logic) and that the identity $\mathsf{i} \in \mathbb{A}$ tracks a map $r\colon \mathbb{A}/R \to \mathbb{A}/\tilde{R}$.

To see that this gives us the reflection, let $a \in \mathbb{A}$ track a map $f\colon \mathbb{A}/R \to \mathbb{A}/T$ where $T \in \mathbb{C}_0$. Then since $\tilde{R} \subseteq a^{-1}T$, this map f factors through $\mathbb{A}/\tilde{R}$ (a tracks a map $\mathbb{A}/a^{-1}T \to \mathbb{A}/T$, hence $\mathbb{A}/\tilde{R} \to \mathbb{A}/T$).

To see that this factorization is unique, let a, a' track maps $\mathbb{A}/\tilde{R} \rightrightarrows \mathbb{A}/T$, and let $E \subseteq \tilde{R}$ be a PER representing their equalizer, as above. Then $E \in \mathbb{C}_0$ and $R \subseteq E$ (because a and a' both track factorizations of f); so $\tilde{R} \subseteq E$, i.e. $E = \tilde{R}$. ■

2.7 Calculating indexed products

The following result lets us avoid a few calculations when we're trying to show that some class of modest sets is closed under arbitrary products (indexed by objects of $\mathcal{E}$). Closure under exponentiation by arbitrary objects of $\mathcal{E}$ will follow, since the exponential X^Y is just a Y-indexed product.

Proposition 2.7.1 *Let* $\mathbb{C}_0$ *be a* $\neg\neg$*-closed subobject of* $\mathbb{M}\boldsymbol{od}_0$*; think of it as defining a full subfibration* $\mathcal{C}$ *of* $\mathcal{M}od$ *(over* $\mathcal{S}ep$*). Then to show that* $\mathcal{C}$ *is closed under indexed products along all arrows in* $\mathcal{S}ep$*, it suffices to show that*

- $\mathcal{C}$ *is an exponential ideal (that is, if* $X \in \mathcal{C}$ *then* $X^Y \in \mathcal{C}$ *for any object* Y*); and*
- *if* $(R_i)_{i \in I}$ *is a family of PERs in* $\mathbb{C}_0$ *indexed by an object* I *in the image of* $\mathcal{S}ets \hookrightarrow \mathcal{E}$ *then their intersection* $\bigcap_{i \in I} R_i \in \mathbb{C}_0$*.*

Note. We repeat the argument that was used in [10]. Actually, the result is really about completeness over $\mathcal{S}ep$; but the result implies closure under indexed products (in a weak sense, cf. [14]) along arbitrary maps of the topos.

Proof. (Sketch; section numbers in this proof refer to [10].) We can conclude from the second condition that $\mathcal{C}$ is closed under indexed products along all maps of $\neg\neg$-sheaves (that is, thought of as fibred over $\mathcal{S}ets$, it's closed under indexed products).

Since $\mathbb{C}_0$ is a $\neg\neg$-closed subobject of a $\neg\neg$-sheaf, and hence itself a $\neg\neg$-sheaf, any map $X \to \mathbb{C}_0$ determines (as in 2.5) a diagram

$$\begin{array}{ccccc} R & \longrightarrow & R' & \longrightarrow & R'' \\ \downarrow & & \downarrow & & \downarrow \\ X & \longrightarrow & sX & \longrightarrow & aX \end{array}$$

(where sX is the reflection of X into $\mathcal{S}ep$, and aX is its $\neg\neg$-sheafification), in which both squares are pullbacks.

Now apply the argument of 2.7. ∎

Of course, checking the hypotheses still involves *some* stuffing around with PERs.

3 Categories of predomains (and domains)

3.1 A dominance

From now on we'll be dealing with the closed term model B of the λ-calculus in which terms having the same Böhm tree are identified. $\mathcal{E}$ will denote the corresponding realizability topos, and $\mathbb{B}$ the assembly constructed from B in the standard way, as described in 2.1.3; we can regard $\mathbb{B}$ as an object of $\mathcal{E}$. By the way, the results in this section also hold for the Böhm tree model (see [3]), and can probably be extended to some other models.

If M is a term, we will write $\mathrm{BT}(M)$ for the Böhm tree of M. It's convenient to speak of a closed term as if it were an element of B. So if M, N are closed terms, when we write $M = N$ we really mean

$$\mathrm{BT}(M) = \mathrm{BT}(N)$$

The symbol Ω will stand for the term $\Omega = (\lambda x.xx)(\lambda x.xx)$ (or any other unsolvable term; all are identified in the model).

[This is an unfortunate clash of notation, since we previously used Ω to denote the object of truth-values of $\mathcal{E}$. But both usages are standard; and anyway, one can always tell from the context whether Ω is a term or an object.]

Definition 3.1.1 *(Cf. 2.5) Let Σ be the B-valued assembly with underlying set $\{\top, \bot\}$ and*

$$|\bot| = \{\Omega\}, \quad |\top| = \{i\}.$$

The idea is that Ω represents divergence and i flags convergence. Recall that Σ is actually a subset of Ω, the object of truth-values: see 2.5.1.

Unlike the dominances defined in [20] and [21], the object Σ is *not* closed under countable (or even binary) joins in Ω. We will not prove this here, but the argument is the same as that used in the proof of proposition 4.1.2 below.

Is Σ a dominance? It is, but the argument is messy, so we postpone the proof to 3.3. Actually, all we need is the following result, which uses the same idea but is cleaner to prove.

Proposition 3.1.2 *Σ is a dominance relative to $\mathcal{S}ep$.*

Proof. We need to prove that $X \subset_\Sigma X$ for any object X, and that a Σ-subset of a Σ-subset is a Σ-subset. The first statement is easily seen to be true; so we simply need to show that if $X \subset_\Sigma Y$ and $Y \subset_\Sigma Z$ then $X \subset_\Sigma Z$.

Suppose M tracks the classifying map $Y \to \Sigma$ of $X \subset_\Sigma Y$, and N tracks the classifying map of $Y \subset_\Sigma Z$; then the classifier of $X \subset_\Sigma Z$ is tracked by

$$\lambda x.Nx(Mx)$$

To see this, observe that for any term P, if NP is unsolvable then so is $NP(MP)$, no matter what MP is.

[Note that this proof is very 'uniform' in character; it can easily be modified to prove the indexed versions of the conditions, which are what we really have to verify. Also, I noticed after presenting this paper that the trick employed here is very similar to that used by Lercher in [17]; [3, lemma 8.4.6] is a more accessible reference.] ∎

It may help the reader to see a more 'internal' version, for modest sets rather than assemblies. The fact that the same argument works should demonstrate that reasoning in the internal language is not so different from reasoning 'externally' in the usual way.

Proposition 3.1.3 *Σ is a dominance relative to $\mathcal{M}od$.*

Proof. The proof relies on the results sketched in 2.5. Let

$$m: X \rightarrowtail Y, n: Y \rightarrowtail Z$$

be Σ-subsets, where $X \cong \mathbb{B}/R, Y \cong \mathbb{B}/S, Z \cong \mathbb{B}/T$. Suppose

$$\begin{aligned}\exists M, N \in \mathbb{A}.\ &\text{“}M \text{ tracks a map } Y \to \Sigma\text{”}\\ &\text{“}M \text{ tracks the classifying map of } m\text{”}\\ &\text{“}N \text{ tracks a map } Z \to \Sigma\text{”}\\ &\text{“}N \text{ tracks the classifying map of } n\text{”}\end{aligned}$$

(It was explained earlier how to state these precisely in the internal language.)

Then if we let $Q = (\lambda yz.\lambda x.zx(yx))$, it's tedious but not difficult to verify that

$$\text{“}QMN \text{ tracks a map } Z \to \Sigma\text{”}$$

and

$$\text{“}QMN \text{ tracks the classifying map of } nm\text{”}$$

∎

A little thought (and familiarity with the internal language of a realizability topos) shows that this really amounts to a proof of a 'uniform' statement of the form

$$\begin{aligned}&\forall R, S, T \in \mathbb{M}\mathbf{od}_0. \forall m \in (\mathbb{B}/S)^{(\mathbb{B}/R)}. \forall n \in (\mathbb{B}/T)^{(\mathbb{B}/S)}.\\ &\quad \text{“}m, n \text{ are monos and display } \Sigma\text{-subsets”} \longrightarrow \text{“}nm \text{ displays a } \Sigma\text{-subset”}\end{aligned}$$

This should give some indication of how the other proofs in this paper, which involve manipulating modest sets and codes for maps between them, can be though of as proofs of statements in the internal language.

3.2 Partial map classifiers

Now to describe the lift of a modest set. Intuitively, if R is a PER then the 'lifted PER' $R_\perp$ is defined by: $\Omega R_\perp \Omega$, and if MRM then $M_\bullet R_\perp M_\bullet$ where $M_\bullet = \lambda z.z\mathsf{i}M$. (The motivation is that i is a 'flag' indicating convergence of a computation; note that a computation might converge and yield the answer Ω, and this is different from diverging.) But we need to be able to formalize this definition in the internal logic; and we want the PER $R_\perp$ to be $\neg\neg$-closed. We need to be a bit more careful.

Definition 3.2.1 *If R is a PER on $\mathbb{B}$, define a PER $R_\perp$ by*

$$M R_\perp N \quad \longleftrightarrow \quad \neg(M = N = \Omega) \to (((M\mathsf{f})R(N\mathsf{f}) \wedge M\boldsymbol{k} = N\boldsymbol{k} = \mathsf{i}))$$

(where $\mathsf{f} = \lambda xy.y$).

It's a simple exercise in intuitionistic logic to verify that this gives us a $\neg\neg$-closed partial equivalence relation (using the fact that R is $\neg\neg$-closed), as we'd like. Note that if we'd tried to define $R_\perp$ by cases—that is, by a formula containing a disjunction—it would not have been $\neg\neg$-closed.

Proposition 3.2.2 *If $Y \cong \mathbb{B}/R$, then $Y_\perp \cong \mathbb{B}/R_\perp$ (where $Y_\perp$ denotes the Σ-partial map classifier relative to $\mathcal{S}ep$).*

Proof. There is a mono $\mathbb{B}/R \rightarrowtail \mathbb{B}/R_\perp$, tracked by $\lambda x.x_\bullet$. In fact $Y \subset_\Sigma \mathbb{B}/R_\perp$; its classifier $\mathbb{B}/R_\perp \to \Sigma$ is tracked by

$$\lambda x.x\mathsf{k}$$

Suppose $X' \subset_\Sigma X$, M tracks a map $X' \to Y$ and N tracks the classifier $X \to \Sigma$ of X'. Then the corresponding map $X \to \mathbb{B}/R_\perp$ is tracked by

$$\lambda x.(Nx)(\lambda z.z(Mx))$$

(and note again that if NP is unsolvable then applying NP to $\lambda z.z(MP)$ will give us something unsolvable).

The converse is easy. ■

Again, there is a more internal proof and a more internal statement quantifying over PERs; we don't bother to write them out this time.

Definition 3.2.3 *Call a PER R* ***canonically focal*** *if $\Omega R \Omega$; call a modest set* ***canonically focal*** *if it's the quotient of $\mathbb{B}$ by a canonically focal PER.*

It follows from the above proposition that lifts of modest sets may be taken to be canonically focal. And 2.7.1 can be used to prove:

Proposition 3.2.4 *The fibration of canonically focal sets is closed under indexed products.*

Since the class of canonically focal PERs isn't closed under isomorphism, a statement like this has to be interpreted with care. What we mean is that if $I \in \mathcal{E}$ is an indexing object,

$$\forall R_i \in \mathbb{M}\mathbf{od}_0.\,((\forall i \in I.\,\Omega R_i \Omega) \;\longrightarrow\; (\exists R \in \mathbb{M}\mathbf{od}_0.\,(\Omega R \Omega) \wedge \mathbb{B}/R = \prod_{i \in I}(\mathbb{B}/R_i)))$$

(and further, if $I \in \mathcal{S}ep$ then there is a map that takes $(R_i)_{i \in I} \mapsto R$, cf. [14]).

Finally we have the following useful lemma. Let $\perp \in X_\perp$ denote 'bottom' element of $X_\perp$ (if $X_\perp \cong \mathbb{B}/R_\perp$ then $\perp$ corresponds to the equivalence class containing Ω).

Lemma 3.2.5 *Let X, Y be modest sets. Then any map $f\colon X \to Y_\perp$ has an extension*

$$f_\perp\colon X_\perp \to Y_\perp$$

preserving $\perp$.

Proof. Suppose $M \in \mathsf{B}$ tracks f; then $\lambda x.(x\mathsf{k})(M(x\mathsf{f}))$ tracks $f_\perp$. ∎

One can show in a similar way that maps $f: X_1 \times \cdots \times X_n \to Y_\perp$ extend to 'strict' maps $f_\perp: (X_1)_\perp \times \cdots \times (X_n)_\perp \to Y_\perp$. For if M tracks f, then

$$\lambda x.(\langle \pi_1 x\mathsf{k}, \ldots, \pi_n x\mathsf{k}\rangle)(M(\langle \pi_1 x\mathsf{f}, \ldots, \pi_n x\mathsf{f}\rangle))$$

tracks $f_\perp$, where π_i and $\langle \cdots \rangle$ are n-ary pairing and projection combinators. (It is an interesting fact, though we won't need to use it, that there is an internal version of this extension property, involving a formula quantified over $n \in \mathbb{N}$; the existence of a suitable λ-term is by no means trivial: see [4].)

3.3 Σ is a dominance

We will sketch a proof that Σ, as defined earlier, is a dominance. The result is not strictly required—only the rather weaker statement 3.1.3 is needed—but it is of some independent interest. It also illustrates a method of proving things in realizability toposes, in which we show that a statement is valid not by arguing within the internal logic but by constructing a realizer for it (see 2.2). Such proofs can be rather technical, but are sometimes unavoidable. However, you may want to skip over this section.

We want to show that Σ is a dominance; in other words, that the formulas

$$\top \in \Sigma$$

and

$$\forall p, q \in \Omega.\, (p \in \Sigma) \wedge (p = \top \to q \in \Sigma) \;\to\; (p \wedge q \in \Sigma)$$

are both valid in $\mathcal{E}$. We explain how to handle the second; the first is easier, and left to the reader.

It may be helpful at this stage to observe that if $p \in \Omega$ is a truth-value, the following three formulas are equivalent:

$$p \qquad\qquad p \leftrightarrow \top \qquad\qquad p = \top$$

Recall from 2.5.1 that $p \in \Sigma$ was equivalent to the formula

$$\exists M \in \mathbb{B}.\, (p = \top \leftrightarrow M = \mathsf{i}) \;\wedge\; (p = \perp \leftrightarrow M = \Omega);$$

So the set of realizers $|p \in \Sigma|$ for this formula is

$$\bigcup_{M \in \mathsf{B}} \Big(|M \in \mathbb{B}| \;\wedge\; (((|p \leftrightarrow \top| \to |M = \mathsf{i}|) \wedge (|M = \mathsf{i}| \to |p \leftrightarrow \top|)) \\ \wedge((|p \leftrightarrow \perp| \to |M = \Omega|) \wedge (|M = \Omega| \to |p \leftrightarrow \perp|)))\Big)$$

Similarly, the set of realizers for the assertion we want to prove is

$$\bigcap_{p,q \in \mathbf{P}\mathsf{B}} |p \in \Omega| \wedge |q \in \Omega| \;\to\; (|p \in \Sigma| \wedge (|p \leftrightarrow \top| \to |q \in \Sigma|) \to |p \wedge q \in \Sigma|)$$

To prove the formula valid, then, we must produce an element of this set; i.e. some very large (closed) λ-expression.

Rather than exhibit this term in its full gruesomeness, we'll give some idea of how it can be constructed by showing how, given $p, q \in \mathbf{P}\mathsf{B}$ and terms

$$P \in |p \in \Sigma|,\ Q \in |p \leftrightarrow \top| \to |q \in \Sigma|$$

we can define a term R using P and Q together with a realizer

$$S \in |R = \mathsf{i}| \to |p \wedge q \leftrightarrow \top|.$$

This essentially tells us how to construct an element of

$$|p \in \Sigma| \wedge (|p \leftrightarrow \top| \to |q \in \Sigma|) \ \to\ \bigcup_R |R \in \mathbb{B}| \wedge (|R = \mathsf{i}| \to |p \wedge q \leftrightarrow \top|)$$

i.e. a realizer for

$$p \in \Sigma \wedge (p = \top \to q \in \Sigma) \ \to\ \exists R \in \mathbb{B}.\, R = \mathsf{i} \to (p \wedge q) = \top.$$

This is at least part of what's required. It's tedious (and not completely trivial!) to fill in the rest, namely to find realizers for

$$\begin{array}{rcl} p \wedge q = \top & \to & R = \mathsf{i} \\ p \wedge q = \bot & \leftrightarrow & R = \Omega \end{array}$$

First a bit of notation: suppose p_0 and p_1 are suitable projection combinators; then if P is a term, we abbreviate $\mathsf{p}_0 P$ and $\mathsf{p}_1 P$ by P_0 and P_1. So, for example, P_{011} denotes $\mathsf{p}_1(\mathsf{p}_1(\mathsf{p}_0 P))$.

Since $P \in |p \in \Sigma|$, P_0 is a term such that P_1 is in the set

$$((|p \leftrightarrow \top| \to |P_0 = \mathsf{i}|) \wedge (|P_0 = \mathsf{i}| \to |p \leftrightarrow \top|)) \wedge \cdots$$

So P_{101} is an element of

$$|P_0 = \mathsf{i}| \to |p \leftrightarrow \top|;$$

that is, if $P_0 = \mathsf{i}$, $P_{101} P_0 \in |p \leftrightarrow \top|$.

Let $T = Q(P_{101} P_0)$. Then if $P_0 = \mathsf{i}$, $T \in |q \in \Sigma|$, i.e. T_0 is a term such that T_1 is in the set

$$((|q \leftrightarrow \top| \to |T_0 = \mathsf{i}|) \wedge (|T_0 = \mathsf{i}| \to |q \leftrightarrow \top|)) \wedge \cdots$$

We let $R = P_0 T_0$. We need to find S.

Now, the formulas

$$\forall C, D \in \mathbb{B}.\, CD = \mathsf{i} \to \neg(C = \Omega)$$

and

$$\forall C, D \in \mathbb{B}.\, (C = \mathsf{i} \wedge CD = \mathsf{i}) \to D = \mathsf{i}$$

are valid in $\mathcal{E}$ (they are valid in $\mathcal{S}ets$ and $\neg\neg$-closed, hence valid in $\mathcal{E}$: see [9, proposition 6.5]). Let U and Y be realizers for them; it doesn't matter exactly which terms we choose, but we could put $U = \mathsf{k}(\mathsf{k}(\mathsf{k}(\mathsf{ki})))$ and $Y = \mathsf{k}(\mathsf{k}(\mathsf{ki}))$, for example.

Also, the formula

$$P_0 = \mathsf{i} \to p = \top$$

is valid; let V be a realizer for it. And lastly,

Lemma 3.3.1 $\neg(P_0 = \Omega) \to P_0 = i$.

Proof. By the definition of P,

$$p = \perp \to P_0 = \Omega$$

is valid; therefore

$$\neg(P_0 = \Omega) \to \neg(p = \perp)$$

is valid. But $\neg(p = \perp)$ is equivalent to $\neg\neg(p = \top)$, and since

$$p = \top \to P_0 = \mathsf{i}$$

is valid,

$$\neg\neg(p = \top) \to \neg\neg(P_0 = \mathsf{i})$$

is valid. But $\neg\neg(P_0 = \mathsf{i})$ is equivalent to $P_0 = \mathsf{i}$. ∎

So we can let W be a realizer for this formula.

Then if $R = \mathsf{i}$, $W(V(UR)) \in |p \leftrightarrow \top|$; that is, $\lambda x.W(V(Ux))$ realizes

$$R = \mathsf{i} \to p = \top.$$

Similarly if $R = \mathsf{i}$, i.e. if $P_0 T_0 = \mathsf{i}$, then $V(UR) \in |P_0 = \mathsf{i}|$ and so $Y\langle V(UR), R\rangle \in |T_0 = \mathsf{i}|$. But T_{101} realizes

$$T_0 = \mathsf{i} \to q = \top$$

and so $\lambda x.T_{101}(Y\langle V(Ux), x\rangle)$ realizes

$$R = \mathsf{i} \to q = \top.$$

But the formula

$$(p = \top \wedge q = \top) \to (p \wedge q) = \top$$

is clearly valid; suppose it's realized by Z. Then

$$\lambda x.Z\langle W(V(Ux)), T_{101}(Y\langle V(Ux), x\rangle)\rangle$$

realizes $R = \mathsf{i} \to (p \wedge q) = \top$.

The fact that this proof is so messy may be disturbing; it probably means that we should look for a better definition of Σ. On the other hand, note that the basic idea is simple (as illustrated in 3.1.2).

3.4 No linked partial order

The synthetic (pre)domains that have been studied in previous papers generally come equipped with an intrinsic ω-complete partial order. This agrees with the expected partial order on (computable) partial function spaces, and hence with the entailment order on $\Sigma = [1 \rightharpoonup 1]$.

For example, if X is one of the 'complete Σ-spaces' in $\mathcal{E}\!f\!f$ (as defined in [20]) one puts

$$x \leq x' \quad \text{iff} \quad \forall \phi \in \Sigma^X.\, \phi(x) \to \phi(x')$$

where Σ is the so-called 'r.e. subobject classifier'. It is not at first clear how to prove anything about this partial order; the key fact is that it is *linked*, which means that

$$x \leq x' \quad \text{iff} \quad \exists! f \in X^\Sigma.\, f(\perp) = x \wedge f(\top) = x'$$

It follows from this that the intrinsic order on function spaces and products coincides with the pointwise order.

The concept of 'linked', though simple, seems to be rather important; it was first isolated in [20], though the method of proving that complete Σ-spaces are linked is adapted from a proof in [30] and hence goes back to classical recursion theory. Note that the fact that the entailment order on Σ is linked is assumed as an axiom in [11].

In the predomains studied in [21], in which Σ is defined differently (in a manner quite similar to the way the object Σ in the present paper is defined) the same definition of the intrinsic order does not seem to work. But one can simply *define*

$$x \leq x' \quad \text{iff} \quad \exists! f \in X^{\Sigma}.\, f(\bot) = x \wedge f(\top) = x'$$

and pick out a nice class of objects for which this is an ω-complete partial order. (It is not immediately obvious that this strategy will work, but it does.)

We will show that this technique can't work here. Finding a *well-behaved* partial order thus seems to be a difficult problem. Luckily, as we'll see, there's no need to look for one; we can do without it.

If our predomains were partially ordered, we would certainly demand that $\bot \leq \top$ in Σ. Suppose that

$$\forall p, p' \in \Sigma.\, p \leq p' \rightarrow \exists f \in \Sigma^{\Sigma}.\, f(\bot) = p \wedge f(\top) = p'$$

Then there would have to be a suitable map $L \times \Sigma \rightarrow \Sigma$, where

$$L = \{\langle p, p'\rangle \,|\, p \leq p'\} \rightarrowtail \Sigma \times \Sigma$$

Unwinding, this means that we must have a term $M \in \mathsf{B}$ such that

$$MPQR \quad = \quad \begin{cases} P & \text{if } R = \Omega \\ Q & \text{if } R = \mathsf{i} \end{cases}$$

where $\langle P, Q\rangle = \langle \Omega, \Omega\rangle, \langle \Omega, \mathsf{i}\rangle$ or $\langle \mathsf{i}, \mathsf{i}\rangle$.

Proposition 3.4.1 *Such a term M cannot exist.*

Proof. This is a simple consequence of [3, theorem 14.4.8]. Suppose M existed. Then $M\Omega\mathsf{i}\Omega = \Omega$ so, using the terminology adopted there, this is caused by either $P = \Omega$ or $R = \Omega$. But $M\mathsf{ii}\Omega = \mathsf{i}$, so it is not caused by $R = \Omega$; and $M\Omega\mathsf{ii} = \mathsf{i}$, so it is not caused by $P = \Omega$. Thus we have a contradiction. ∎

The (vague) intuition is that the sequential nature of computation in B prevents any reasonable partial order from being linked. Note that the same argument applies to any choice of Σ isomorphic to ours, and in fact to a rather larger class of prospective Σ. Of course, this does not prove that there is no well-behaved partial order at all, since the notion of 'well-behaved' is not a precise one.

The proposition shows that the predomains we will define cannot fit into the general framework of Hyland's described in [11]. Hyland is, in fact, developing a 'stable' version of that theory for which the predomains of this paper ought to provide a model. However, this is still very much work in progress. In any case, we can proceed in a more concrete fashion, as we'll see.

Note. One might hope that there is an intrinsic order which coincides with the stable order on functions spaces; indeed, the conference version of this paper included a section on stability properties of our domains. But the situation remains unclear. For example, by the proof of [15, theorem 8], the elements of

$$2_\perp{}^{2_\perp{}^{2_\perp}}$$

in our model cannot coincide with the elements of

$$[[\mathrm{Bool} \to \mathrm{Bool}] \to \mathrm{Bool}]$$

in the classical stable model. This is because the stable domain contains a 'true-separator', whereas such a thing is not syntactically definable.

3.5 A kind of chain completeness

Since we have no good partial order to work with, we have to define 'chain completeness' in a more concrete way, looking at the PERs themselves. In fact, the definition turns out to be rather natural.

We will constantly be using the fact that 'application is continuous with respect to the Böhm tree topology': that is, if M is any term and N_n, N are terms with

$$\mathrm{BT}(N_1) \subseteq \mathrm{BT}(N_2) \subseteq \mathrm{BT}(N_3) \ldots$$

and

$$\mathrm{BT}(N) = \bigcup_n \mathrm{BT}(N_n)$$

then

$$\mathrm{BT}(MN) = \bigcup_n \mathrm{BT}(MN_n).$$

(This is actually a special case of a more general theorem; see [3, chapter 19]). We will abuse notation and write

$$M(\bigcup_n N_n) = \bigcup_n MN_n.$$

Now, let c_n denote the nth Church numeral and define

$$\ulcorner n \urcorner \;=\; \lambda f.\mathsf{c}_n f\Omega;$$

that is,

$$\ulcorner 0 \urcorner = \lambda f.\Omega,\ \ulcorner 1 \urcorner = \lambda f.f\Omega,\ \ulcorner 2 \urcorner = \lambda f.f(f\Omega), \ldots$$

Observe that, in a sense, the terms $\ulcorner n \urcorner$ form a sequence of approximations to the fixed point combinator

$$\mathsf{y} \;=\; \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))\,;$$

we say this because

$$\mathrm{BT}(\mathsf{y}) = \bigcup_n \mathrm{BT}(\ulcorner n \urcorner).$$

Definition 3.5.1 *A PER R on $\mathbb{A}$ is* ***complete*** *if the following statement is valid:*

$$(\forall M, N \in \mathbb{B}.\ \forall n.\,(M\ulcorner n \urcorner)R(N\ulcorner n \urcorner)) \;\longrightarrow\; (M\mathsf{y})R(N\mathsf{y}).$$

And a modest set is ***complete*** *if it's the quotient of $\mathbb{B}$ by a complete PER.*

For example the normal definitions of $2, \mathbb{N}, \mathbb{N}^{\mathbb{N}}, \ldots$ are complete PERs. We will see that complete PERs are closed under (total) function spaces, lifting (and hence Σ-partial function spaces), datatype constructions, polymorphic types and so on. In fact, all of these except for lifting follow from the following simple result.

Proposition 3.5.2 *The category of complete modest sets is reflective.*

Proof. Closure under intersection is obvious; and it's straightforward to show, using the fact that application is continuous, that the other hypotheses of 2.6.1 also hold. ∎

There is one particularly interesting example of a complete PER. Let ω be the B-valued assembly with underlying set $\{0, 1, 2, \ldots\}$ and put

$$|n| = \{\ulcorner n \urcorner\};$$

and let $\bar{\omega}$ be the B-valued assembly with underlying set $\{0, 1, 2, \ldots, \infty\}$ and put

$$|n| = \{\ulcorner n \urcorner\},\ |\infty| = \{\mathsf{y}\};$$

there is an obvious mono $\omega \rightarrowtail \bar{\omega}$.

Note that any map from ω into a complete modest set extends uniquely to $\bar{\omega}$; in fact, the defining condition for complete PERs tells us that if M tracks a map $\omega \to X$ then it also tracks the extension $\bar{\omega} \to X$.

We have a continuity result which is easy but important: Any map between complete modest sets 'preserves sups of chains'; more precisely,

Proposition 3.5.3 *Let X and Y be complete modest sets, and let $f\colon X \to Y$. If $g\colon \omega \to X$ extends (uniquely) to $h\colon \bar{\omega} \to X$, then the unique extension of $fg\colon \omega \to Y$ to a map $\bar{\omega} \to Y$ is just fh.*

Proof. By the uniqueness of extensions, it suffices to show that fh extends fg; that is, that they agree on ω. But they do, because h extends g. ∎

And we do have sups of enough chains to be able to calculate fixed points (which, after all, was the main reason why we wanted 'chain completeness').

Proposition 3.5.4 *Suppose X is complete and canonically focal; then any map $X \to X$ has a fixed point. In fact, there is a fixed point operator $X^X \to X$.*

Remark. In other words, X has the so-called *fixed point property*. This property was first studied by Lawvere, and many years later Scott suggested that it could prove useful in an axiomatic treatment of domain theory.

Proof. Suppose we have a map $X \to X$ tracked by a term P. Define $\omega \to X$ to be the map tracked by the term

$$M = \lambda x.xP;$$

this extends to a map $\bar{\omega} \to X$.

By the definition of complete PER, M tracks the extended map $\bar{\omega} \to X$. But $M\mathsf{y} = \mathsf{y}P$ is a code for a fixed point of the map $X \to X$ (indeed, a fixed point 'on the nose' for codes).

The internal version—that there exists a fixed point operator $X^X \to X$, tracked by y—follows from this argument by adding some λ's. ∎

Note the intensional nature of this proof; it has a rather different flavour from the usual fixed-point theorem for domains.

3.6 Closure under lifting

We would like to use the category of complete PERs (identified with the complete modest sets) as our category of predomains, and the complete, canonically focal PERs as our category of domains.

For this category to be useful as a category of predomains, it had better be closed under the lift construction so that it contains Σ-partial function spaces as well as total ones. So this is the last thing we need to check. The idea is very simple (cf. citePh4) and there is less to the proof than meets the eye.

First we need a lemma that says we are allowed to prove this fact 'externally', and another that states the—external—continuity of application 'the other way round' (there's an internal counterpart, of course, but we won't need it).

Lemma 3.6.1 *Let $\mathcal{C}$ be a class of PERs, corresponding to a $\neg\neg$-closed predicate ϕ on $\mathbb{M}\boldsymbol{od}_0$ (the internal object of $\neg\neg$-closed PERs, which is a $\neg\neg$-sheaf). That is, $\mathcal{C}$ comes from an ordinary subset of the set of PERs on B. In order to prove that*

$$\forall R \in \mathbb{M}\boldsymbol{od}_0.\, \phi(R)$$

is valid, it suffices to prove the external fact that $R \in \mathcal{C}$ for all PERs R.

Proof. This fact is immediate given the description of internal logic of realizability toposes in [9]. Since $\mathbb{M}\mathbf{od}_0$ is a $\neg\neg$-sheaf, the formula will be valid if and only if

$$\bigcap_R |\phi(R)| \neq \varnothing$$

But since ϕ is the $\neg\neg$-closed predicate corresponding to $\mathcal{C}$,

$$\phi(R) = \begin{cases} \mathsf{B} & \text{if } R \in \mathcal{C}; \\ \varnothing & \text{otherwise.} \end{cases}$$

So the intersection is nonempty precisely when $R \in \mathcal{C}$ for all R. ∎

Lemma 3.6.2 *If $M = \bigcup_n M_n$ then $MN = \bigcup_n M_n N$.*

Proof. Define $\mathsf{a} = \lambda x f. fx$. Then

$$MN = \mathsf{a}NM = \mathsf{a}N(\bigcup_n M_n) = \bigcup_n \mathsf{a}NM_n = \bigcup_n M_n N$$

by continuity of application. Alternatively, this result is a corollary of 3.5.3. ∎

Proposition 3.6.3 *The category of complete PERs is closed under lifting.*

Proof. Let $\mathcal{C}$ be the collection of PERs R on B such that

$$\text{if } R \text{ is complete, so is } R_\perp$$

(where $R_\perp$ is the PER defined in section 3.1; we are taking the external analogues of all the internal definitions). It corresponds to a certain predicate ϕ on $\mathbb{M}\mathbf{od}_0$ of the form

$$R \text{ complete } \longrightarrow R_\perp \text{ complete}$$

which is $\neg\neg$-closed since the formulas on either side of the implication symbol are $\neg\neg$-closed.

The proposition is equivalent to the validity of

$$\forall R \in \mathbb{M}\mathbf{od}_0.\, \phi(R)$$

and so, by the first lemma, it suffices to prove the external fact that all PERs R on B belong to $\mathcal{C}$; that is: whenever R is a complete PER, then so is $R_\perp$.

First some preliminary definitions. For each $i \in \mathbb{N}$ let

$$\mathsf{s}_i = \lambda f. f^i(\ulcorner n \urcorner f);$$

then $\mathsf{s}_i \ulcorner n \urcorner = \ulcorner n + i \urcorner$ for all n. Now if M is any term let

$$M^{(i)} = \lambda x. M(\mathsf{s}_i x);$$

then $M^{(i)} \ulcorner n \urcorner = M \ulcorner n + i \urcorner$ for all n. Observe that

$$M\mathsf{y} = \bigcup_n M \ulcorner n \urcorner = \bigcup_n M \ulcorner n + i \urcorner = \bigcup_n M^{(i)} \ulcorner n \urcorner = M^{(i)}\mathsf{y}.$$

Now: Let R be a complete PER and suppose we have terms M, N such that for all n, $(M\ulcorner n \urcorner) R_\perp (N \ulcorner n \urcorner)$; we want to show that $(M\mathsf{y}) R_\perp (N\mathsf{y})$. By the definition of $R_\perp$, for all n we have that if $\neg(M\ulcorner n \urcorner = N \ulcorner n \urcorner = \Omega)$ then

$$(M\ulcorner n \urcorner \mathsf{f}) R (N \ulcorner n \urcorner \mathsf{f}) \wedge M\ulcorner n \urcorner \mathsf{k} = N \ulcorner n \urcorner \mathsf{k} = \mathsf{i}.$$

Suppose that $\neg(M\mathsf{y} = N\mathsf{y} = \Omega)$. Since $M\mathsf{y} = \bigcup_n M\ulcorner n \urcorner$ and $N\mathsf{y} = \bigcup_n N \ulcorner n \urcorner$, we must have

$$\neg(M\ulcorner i \urcorner = N \ulcorner i \urcorner = \Omega)$$

for some i sufficiently large. Then for all n, we have

$$\neg(M^{(i)}\ulcorner n \urcorner = N^{(i)} \ulcorner n \urcorner = \Omega)$$

and hence

$$(M^{(i)}\ulcorner n \urcorner \mathsf{f}) R (N^{(i)} \ulcorner n \urcorner \mathsf{f});$$

this shows that $(M^{(i)}\mathsf{yf}) R (N^{(i)}\mathsf{yf})$ and hence $(M\mathsf{y}) R (N\mathsf{y})$. Also,

$$M^{(i)}\ulcorner n \urcorner \mathsf{k} = N^{(i)} \ulcorner n \urcorner \mathsf{k} = \mathsf{i},$$

and hence

$$M\mathsf{yk} = M(\bigcup_n \ulcorner n \urcorner)\mathsf{k} = (\bigcup_n M\ulcorner n \urcorner)\mathsf{k} = \bigcup_n M\ulcorner n \urcorner \mathsf{k} = \bigcup_n M^{(i)}\ulcorner n \urcorner \mathsf{k} = \bigcup_n \mathsf{i} = \mathsf{i};$$

and so $M\mathsf{yk} = \mathsf{i}$. Similarly for N. So we are done. ■

It would be nice to have a proof entirely in the internal language (as in [20]); but I haven't been able to find one.

3.7 Fixed point induction

There is one more ingredient we need to make the theory usable. Let X be complete and canonically focal; then by 3.5.4 there is a fixed point operator $y: X^X \to X$. To reason about fixed points, we have to be able to invoke the rule

$$\frac{\phi(\bot) \qquad \phi(x) \to \phi(fx)}{\phi(yf)}$$

Definition 3.7.1 *A predicate $\phi(x)$ is* ***admissible*** *if this rule is admissible for ϕ.*

What are the admissible predicates? As no partial order has been defined on X, the notion of 'inclusive predicate' will be of no use here.

A **Σ-predicate** on X is a map $\phi: X \to \Sigma$. The Σ-predicates are a reasonably large and useful class of properties of X: they include all the decidable predicates, of course.

Proposition 3.7.2 *If $\phi(x)$ is a Σ-predicate, both $\phi(x)$ and $\neg\phi(x)$ are admissible.*

Proof. Let $X \cong \mathbb{B}/R$ and let N track $\phi: \mathbb{B}/R \to \Sigma$; let P track $f: X \to X$. Then $\mathsf{y}P \in |yf \in X|$. More precisely, since we are working internally, the R-equivalence class of $\mathsf{y}P$ corresponds to $x \in X$ under the isomorphism $X \cong \mathbb{B}/R$.

If $M \in |x \in X|$, $\phi(x)$ holds iff $NM = \mathsf{i}$. The hypotheses of the rule tell us that $N\Omega = \mathsf{i}$ and that if $NM = \mathsf{i}$ then $N(PM) = \mathsf{i}$. So for all n, $N(\ulcorner n \urcorner P) = \mathsf{i}$. Hence by continuity, $N(\mathsf{y}P) = \mathsf{i}$.

The argument for $\neg\phi(x)$ is similar. ■

In fact a rather larger class of predicates is admissible: for example, if $\phi(x, y)$ is an admissible predicate on $X \times Y$ then $\forall y.\phi(x, y)$ is an admissible predicate on X. For example, if $X = [Y \rightharpoonup Z]$ we have, for each $y \in Y$, a Σ-predicate $\phi(f, y)$: '$f(y)\downarrow$'. Hence the predicates

$$\begin{aligned} \text{'}f\text{ is total'} &= \forall y.\phi(f, y) \\ \text{'}f\text{ is nowhere defined'} &= \forall y.\neg\phi(f, y) \end{aligned}$$

are both admissible.

However, this still seems less general than 'inclusive'. There is clearly room for further investigation. For instance, it is not clear when a predicate of the form $\exists y.\phi(x, y)$ is admissible.

4 Interpreting PCF

4.1 A model of call-by-name PCF

It's not at first obvious that the domains we've defined are usable; we're used to working with posets, and they're not posets. For example, how can we use them to interpret PCF sensibly? In fact this is a very straightforward exercise—and does not use the full power of our results, such as the ability to model polymorphic types—but it already raises some interesting questions. For example, it gives some indication of how our domains are related to operational semantics, though this relationship is still poorly understood.

All these results can be extended to polymorphic versions of PCF without much extra trouble (cf. the remarks at the end of [24]). Unfortunately, we would then need an adequacy theorem for 'classical' domain-theoretic models of polymorphic PCF; Coquand has proved a suitable result, but this work remains unpublished.

$$\begin{array}{rcll}
\texttt{tt}, \texttt{ff} & : & o & \\
\underline{n} & : & \iota & \text{for } n = 0, 1, 2, \ldots \\
\texttt{succ}, \texttt{pred} & : & \iota \to \iota & \\
\texttt{zero?} & : & \iota \to o & \\
\texttt{cond}_o & : & o \to o \to o \to o & \\
\texttt{cond}_\iota & : & o \to \iota \to \iota \to \iota & \\
\texttt{Y}^\sigma & : & (\sigma \to \sigma) \to \sigma & \text{for each type } \sigma
\end{array}$$

Figure 1: Constants of PCF

$$\begin{array}{rcl}
\texttt{cond tt } x \ y & \to & x \\
\texttt{cond ff } x \ y & \to & y \\
\texttt{zero? } \underline{0} & \to & \texttt{tt} \\
\texttt{zero? } \underline{n+1} & \to & \texttt{ff} \\
\texttt{succ } \underline{n} & \to & \underline{n+1} \\
\texttt{pred } \underline{0} & \to & \underline{0} \\
\texttt{pred } \underline{n+1} & \to & \underline{n} \\
\texttt{Y} f & \to & f(\texttt{Y} f)
\end{array}$$

Figure 2: Rewrite rules of PCF

We'll only be concerned with call-by-name PCF here; presumably everything also works for call-by-value PCF, but will be somewhat messier (for reasons that will emerge soon). We only give a brief, informal description of PCF here; more details may be found in [27], and many other papers. In particular, problems such as keeping track of contexts, which have to be attended to in a formal presentation, are glossed over here.

PCF is a simply typed λ-calculus with ground types ι (integers) and o (booleans), and suitable constants with assiciated rewrites rules. We let $s, t, \ldots$ denote PCF terms; if we want to emphasize that t is of type σ, we write $t : \sigma$.

To model this, we must first give an interpretation of the types. Let

$$\begin{array}{rcl}
[\![\iota]\!] & = & \mathbb{N}_\perp \\
[\![o]\!] & = & 2_\perp \\
[\![\sigma \to \tau]\!] & = & [\![\tau]\!]^{[\![\sigma]\!]}
\end{array}$$

Now for the terms. Suppose $t : \tau$ is a PCF term with free variables $x_1 : \sigma_1, \ldots, x_n : \sigma_n$. We want to define

$$[\![t]\!] \colon [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to [\![\tau]\!].$$

We can do this in the standard way once we know how to interpret the constants; but this is easy. Note that a constant of type τ will denote a map $1 \to [\![\tau]\!]$. So, for example, for $[\![\underline{n}]\!]$ we simply take the appropriate map $1 \to \mathbb{N}$ and compose with $\mathbb{N} \rightarrowtail \mathbb{N}_\perp$ to get

$$1 \to \mathbb{N}_\perp;$$

and to obtain $[\![\texttt{cond}_\iota]\!]$ we start with the usual map $2 \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, which gives us

$$2 \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}_\perp;$$

we then apply (the remark after) lemma 3.2.5 to get a map

$$2_\perp \times \mathbb{N}_\perp \times \mathbb{N}_\perp \to \mathbb{N}_\perp$$

and curry to get the desired map

$$1 \to ((\mathbb{N}_\perp{}^{\mathbb{N}_\perp})^{\mathbb{N}_\perp})^{2_\perp} = [\![o \to \iota \to \iota \to \iota]\!].$$

Apart from $\mathtt{Y}^\sigma$, the other constants are similar. As for $\mathtt{Y}^\sigma$: note that for any type σ, $[\![\sigma]\!]$ is complete and canonically focal; so by 3.5.4, we have a fixed point operator

$$[\![\sigma]\!]^{[\![\sigma]\!]} \to [\![\sigma]\!]$$

(tracked by y); and this gives us the desired map

$$[\![\mathtt{Y}^\sigma]\!]: 1 \to [\![\sigma]\!]^{([\![\sigma]\!]^{[\![\sigma]\!]})} = [\![(\sigma \to \sigma) \to \sigma]\!].$$

It's easy to check that this interpretion has the following property: if $t : \iota$ is a closed term then $t \twoheadrightarrow \underline{n}$ implies $[\![t]\!] = n$. In fact, we have

Proposition 4.1.1 (adequacy) *$t \twoheadrightarrow \underline{n}$ if and only if $[\![t]\!] = n$.*

Note that it's not obvious how to go about proving this; the method of [27] doesn't seem to apply here, as our 'domains' aren't partially ordered. We'll see in a minute that the idea of the proof is to use proposition 4.2.1 to reduce it to the result proved in [24], and hence ultimately to the Plotkin/Sazonov result.

First, there is a rather tantalising set of questions suggested by the following simple observation: cf. the example on pp. 380–81 of [3].

Proposition 4.1.2 *There is no 'parallel-or' map $2_\perp \times 2_\perp \to 2_\perp$.*

Proof. Let $\mathsf{t} = \lambda xy.x$ and $\mathsf{f} = \lambda xy.y$. Then we can describe $2_\perp$ as the B-valued assembly with underlying set $\{\perp, tt, ff\}$ and

$$|\perp| = \{\Omega\}, \quad |tt| = \{\mathsf{t}_\bullet\}, \quad |ff| = \{\mathsf{f}_\bullet\}.$$

Also, let $\langle M, N\rangle$ denote $\mathsf{p}MN$ where p is a suitable pairing combinator. Finally, we let $M \cap N$ denote 'the' term such that $\mathrm{BT}(M \cap N) = \mathrm{BT}(M) \cap \mathrm{BT}(N)$. This is harmless since such a term must exist, and is unique up to Böhm tree equality, and so defines a unique element of B.

Suppose a parallel-or map existed, and let P track it. By assumption,

$$P\langle \mathsf{t}_\bullet, \Omega\rangle = P\langle \Omega, \mathsf{t}_\bullet\rangle = \mathsf{t}_\bullet;$$

hence

$$P\langle \mathsf{t}_\bullet, \Omega\rangle \cap P\langle \Omega, \mathsf{t}_\bullet\rangle = \mathsf{t}_\bullet.$$

But by the stability of application [3, corollary 14.4.11],

$$\begin{array}{ll} P\langle \mathsf{t}_\bullet, \Omega\rangle \cap P\langle \Omega, \mathsf{t}_\bullet\rangle & \\ \quad = & P(\langle \mathsf{t}_\bullet, \Omega\rangle \cap \langle \Omega, \mathsf{t}_\bullet\rangle) \\ \quad = & P\langle \Omega, \Omega\rangle \\ \quad = & \Omega \end{array}$$

which is a contradiction. ∎

In fact, there are no obvious counterexamples to full abstraction or even universality. Clearly these problems deserve further investigation. We will soon see a 'fake' full abstraction result which is suggestive but not, unfortunately, very deep.

$$\begin{array}{rcl}
\Lambda(x) & = & x \\
\Lambda(st) & = & \Lambda(s)\Lambda(t) \\
\Lambda(\lambda x.t) & = & \lambda x.\Lambda(t) \\
\\
\Lambda(\mathtt{tt}) & = & \lambda xy.x \\
\Lambda(\mathtt{ff}) & = & \lambda xy.y \\
\Lambda(\underline{n}) & = & \mathsf{c}_n \\
\Lambda(\mathtt{succ}) & = & \lambda y.\lambda fx.f(yfx) \\
\Lambda(\mathtt{pred}) & = & \pi \\
\Lambda(\mathtt{cond}) & = & \lambda xyz.xyz \\
\Lambda(\mathtt{Y}^\sigma) & = & \Theta
\end{array}$$

Figure 3: Translation of PCF into the untyped λ-calculus

4.2 Relationship with the untyped λ-calculus

Here we'll describe why the adequacy result has something to do with the 'adequacy' of a translation of PCF into the untyped λ-calculus.

For any PCF term t we can define an untyped λ-term $\Lambda(t)$ as sketched in the figure. Here c_n denotes the nth Church numeral, $\Theta = (\lambda xy.y(xxy))(\lambda xy.y(xxy))$ is Turing's fixed point combinator, and π is a predecessor combinator

$$\lambda uft.(t(\lambda p.pu(\lambda fx.x))(\lambda zq.q(f(z(\lambda xy.x))(z(\lambda xy.y)))(\lambda gv.g((z(\lambda xy.y))gv)))(\lambda xy.x)$$

The translation is *faithful* in the following sense:

$$t \twoheadrightarrow_N t' \qquad \text{implies} \qquad \Lambda(t) \twoheadrightarrow_\beta \Lambda(t')$$

Observe that t is a closed term if and only if $\Lambda(t)$ is closed. Suppose $t : \iota$ is closed; then $t \twoheadrightarrow_N \underline{n}$ implies $\Lambda(t) \twoheadrightarrow_\beta \mathsf{c}_n$. Slightly more surprisingly,

Proposition 4.2.1 (see [24]) *$t \twoheadrightarrow \underline{n}$ if and only if $\Lambda(t) \twoheadrightarrow_\beta \mathsf{c}_n$.*

The proof uses the domains constructed in [21] to reduce the statement to Plotkin's adequacy theorem: [27]. We will explain here why the proposition implies the adequacy result 4.1.1.

We will need an alternative description of the partial map classifiers of $\mathbb{N}$ and 2. Recall from 2.1.5 that any subset $C \subseteq \mathsf{B}$ gives rise to a $\neg\neg$-closed subobject of $\mathbb{B}$. Define two $\neg\neg$-closed subobjects of $\mathbb{B}$ as follows:

$$\begin{array}{rl}
I \rightarrowtail \mathbb{B} & \text{corresponds to} \quad \{\Omega, \mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2, \ldots\} \subset \mathsf{B} \\
O \rightarrowtail \mathbb{B} & \text{corresponds to} \quad \{\Omega, \mathsf{t}, \mathsf{f}\} \subset \mathsf{B}
\end{array}$$

Lemma 4.2.2 *$I \cong \mathbb{N}_\perp$ and $O \cong 2_\perp$.*

Proof. Recall the earlier descriptions of $\mathbb{N}_\perp$ and $2_\perp$. The term $\lambda y.y\mathsf{i}y_\bullet$ tracks a map $I \to \mathbb{N}_\perp$, and the term $\lambda x.x\mathsf{f}$ tracks a map $\mathbb{N}_\perp \to I$; and these two maps are mutually inverse. The proof for $2_\perp$ is similar. ∎

Note that we can think of I and O as being assemblies. So if we choose $[\![\iota]\!] = I$, $[\![o]\!] = O$, the interpretation of PCF we defined previously gives us an interpretation of PCF in the category of assemblies.

Lemma 4.2.3 *Let $t : \tau$ be any PCF term with free variables $x_1 : \sigma_1, \ldots, x_n : \sigma_n$, so that*

$$[\![t]\!] \colon [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to [\![\tau]\!];$$

think of this as a map between assemblies; let f be the corresponding function between the underlying sets. For all $u_1 \in [\![\sigma_1]\!], \ldots, u_n \in [\![\sigma_n]\!]$ (i.e. elements of the underlying sets), if we are given any

$$M_1 \in |u_1 \in [\![\sigma_1]\!]|, \ldots, M_n \in |u_n \in [\![\sigma_n]\!]|$$

then

$$(\Lambda(t))[x_1 := M_1, \ldots, x_n := M_n] \in |f(u_1, \ldots, u_n) \in [\![\tau]\!]|.$$

The proof proceeds by induction, and is similar to the main lemma of [24].

In particular, let t be a *closed* term, so that $[\![t]\!] \colon 1 \to [\![\tau]\!]$. Regarding $[\![\tau]\!]$ as an assembly, $[\![t]\!]$ corresponds to an element $u \in [\![\tau]\!]$ (the underlying set of the assembly); and then

$$\Lambda(t) \in |u \in [\![\tau]\!]|.$$

Using this fact, it's easy to see why 4.1.1 follows from 4.2.1.

Also, as promised earlier, we can state a 'fake' full abstraction theorem. Remember that '$M = N$' here denotes Böhm tree equality (equality in B).

Definition 4.2.4 *If M is an (untyped) term, call a context $C[\cdot]$ a* ***Σ-context for*** *M if either $C[M] = \mathsf{i}$ or $C[M] = \Omega$. And if M, N are terms, call M and N* ***Σ-equivalent*** *if for all contexts $C[\cdot]$ that are Σ-contexts for both M and N, $C[M] = \mathsf{i}$ iff $C[N] = \mathsf{i}$.*

This is a stronger property than we actually need, but it's easy to state. It should be compared with the notion of *solvable equivalence* introduced for technical reasons by Wadsworth: [3, 15.2.3]. It is in a sense more 'uniform'.

Proposition 4.2.5 *Let s and t be two PCF terms. If their translations $\Lambda(s)$ and $\Lambda(t)$ are Σ-equivalent, then $[\![s]\!] = [\![t]\!]$.*

In other words, if the *untyped translations* of two PCF terms are 'observationally equivalent', then they denote the same element in the model. There ought to be some analogy with the relation of applicative bisimulation defined in [2], but this has not been worked out. Also, if we could show that the translation $t \mapsto \Lambda(t)$ was *abstraction-preserving* in the sense of [19], we could deduce full abstraction from 4.2.5; but there is no reason to expect this.

The proof of 4.2.5 uses methods from [20] to show that the category of 'Σ-spaces' (defined analogously to those of [20]) contains 2 and $\mathbb{N}$, and is closed under exponentials and lifting. As we mentioned earlier, the proposition is not deep. And it is probably not the kind of result we're really looking for: firstly because we have to talk about untyped translations rather than PCF terms themselves, and secondly because the notion of Σ-equivalence of untyped terms doesn't seem very tractable.

Lastly, a curiosity: the methods used here also suggest a way of translating call-by-*value* PCF into the untyped call-by-*name* λ-calculus. Again, this relies on the same 'jamming factor' trick that goes back to Lercher.

5 Final remarks

Perhaps the first thing to say is that the results described above are only a beginning. We are far from understanding any of the categories of domains that live in various realizability toposes, and when we do, it seems likely that they will be understood in quite a different way from the way we see them now. The category described in this paper is particularly mysterious, because it does not seem to fit in well with the 'synthetic' point of view underlying the programme of Hyland begun in [11].

That said, there are a couple of more concrete matters worth raising.

On the interpretation of polymorphic types The predicate that carves the objects-of-objects of the category of 'predomains' (complete PERs) out of $\mathbb{M}\mathbf{od}_0$ are both $\neg\neg$-closed, so *the object of complete PERs and the object of complete, canonically focal PERs are both in the image of the inclusion* $\mathcal{S}ets \hookrightarrow \mathcal{E}$.

That is, because of the 'canonical' nature of the sups of chains and bottoms that are required to exist, the underlying objects of these categories can be thought of as an ordinary set in the usual sense. Some other authors (e.g. the authors of [8]) would call them 'uniformizable' objects. In particular, we can calculate products indexed by these objects simply by taking the intersection of the given collection of PERs, see [10].

This is, of course, just the usual interpretation of polymorphic types in PER models. The point is that for some categories of domains—such as the complete Σ-spaces of [20]—the object-of-objects is not an ordinary set and so Moggi-Hyland polymorphic types (see 2.3), while they exist, cannot be calculated simply as intersections—as they can if we are working with the whole category of modest sets. This has something to do with the fact that while the definition of complete Σ-space implicitly includes some intensional information (the existence of a realizer that computes suprema of chains) the definition of complete PER does not. Complete PERs could be compared with the complete extensional PERs of [8] in this respect; in both cases, we gain this nice property, but pay a price: the categories are not closed under isomorphism, and the definitions are not so clean.

(For another approach to this problem, applicable to arbitrary reflective subcategories of modest sets, see the remarks on the 'uniformized interpretation' in [23].)

Applications of completeness We have not mentioned the practical consequences of the fact that our predomains form a small complete category (cf. [20], [21]). For example, we can interpret

- polymorphic types, in various ways;
- 'covariant' recursive types;
- subtypes and bounded quantification: see [25];
- nondeterminism, via convex powerdomains (i.e. free semilattices).

These constructions have not been analysed in detail. There are problems: for example, we might like a description of the convex powerdomain construction that preserved the property of being canonically focal, but there is no obvious candidate.

Other questions

- *What happened to the partial order?*

In contrast to the situation in [20] and [21], we did not make use of partial order structures in this paper: see 3.4. Since there is a partial order on the underlying model—the specialization order arising from the tree topology—it may be possible to define a well-behaved partial order on certain PERs. But our results show that if it *is* possible, new ideas will certainly be required; and one might also argue that it is not really necessary.

- *What about solving mixed variance domain equations?*

One would like to be able to solve these using the inverse limit method; but due to the absence of partial orders on our predomains, this at first seems to be impossible. Luckily this is not so. In fact we can still use the same *idea* to construct solutions to all the usual domain equations, and hence model any syntactically definable recursive type; and these recursive types can be described explicitly as PERs. As an added bonus, they can be constructed in such a way as to provide us with a model for systems with subtyping and recursive types (such as Quest). However, the details are syntactically messy. They will appear elsewhere.

- *Are we looking at the right model?*

The particular model B we chose to look at is not the only possibility. It might be more natural to use Nakajima trees ('infinite η-expansions' of Böhm trees) rather than Böhm trees (see [3, exercise 19.4.4]); then η-equivalence would be incorporated into our notion of equality on terms. But we would have to reprove the continuity theorems, and so on; this should be possible, but might be tedious.

It would also be interesting to look at the lazy λ-calculus, and at various combinator algebras, to try to obtain categories of domains.

- *How can we relate different models?*

At the moment, there are no good techniques for relating categories of domains in different toposes (the methods used in [24] don't really count). This is a serious gap in the present theory of realizability toposes.

Acknowledgements The work described here formed part of a Ph.D. dissertation written under the supervision of Martin Hyland; I must thank him for being so patient and helpful. I also benefited from discussions and correspondence with many other people.

The conference version of this paper was prepared with the support of the Australian Research Council. I am very grateful to the organisers of TACS '91 for giving me the chance to present it there; many improvements resulted. The final version was written at the LFCS, while I was supported by an 1851 Research Fellowship. Both the substance and the presentation of this paper have benefited greatly from the comments of two anonymous referees.

References

[1] M. Abadi, G. Plotkin: A PER model of polymorphism and recursive types, in: Proc. of 5th Annual Symposium on Logic in Computer Science, 1990.

[2] S. Abramsky: The lazy lambda-calculus, in *Research Topics in Functional Programming* (ed. D. L. Turner), Addison-Wesley, 1989.

[3] H. P. Barendregt: *The Lambda-Calculus: Its Syntax and Semantics (revised edition)*, North-Holland, 1984.

[4] H. P. Barendregt: Self-interpretation in lambda calculus, *J. Functional Programming* **1** (2) (1991) 229–234.

[5] M. Beeson: *Foundations of Constructive Mathematics*, Springer, 1985.

[6] A. Carboni, P. J. Freyd, A. Scedrov: A categorical approach to realizability and polymorphic types, in *Mathematical Foundations of Programming Language Semantics* (eds. M. Main, A. Melton, M. Mislove, D. Schmidt), Springer LNCS 298, 1988.

[7] T. Coquand, G. Huet: Constructions: a higher-order proof system for mechanizing mathematics, in: *EUROCAL '85: European Conference on Computer Algebra*, Springer, 1985.

[8] P. Freyd, P. Mulry, G. Rosolini, D.S. Scott: Extensional PERs, in: Proc. of 5th Annual Symposium on Logic in Computer Science, 1990.

[9] J.M.E. Hyland: The effective topos, in: *The L.E.J. Brouwer Centenary Symposium* (ed. A. S. Troelstra, D. van Dalen), North-Holland, 1982.

[10] J.M.E. Hyland: A small complete category, *Annals of Pure and Applied Logic* 40 (1988) 135-165.

[11] J.M.E. Hyland: First steps towards synthetic domain theory, in: *Category Theory, Proceedings, Como 1990* (eds. A. Carboni, M. C. Pedicchio, G. Rosolini), Springer LNCS 1488, 1991.

[12] J. M. E. Hyland, P. T. Johnstone, A. M. Pitts: Tripos theory, *Math. Proc. Camb. Phil. Soc.* (1980), **88**, 205.

[13] J.M.E. Hyland, A.M. Pitts: The theory of constructions, categorical semantics and topos-theoretic models, in *Categories in Computer Science and Logic (Proc. Boulder 1987), Contemp. Math.* **92** (1989) 137–199.

[14] J.M.E. Hyland, E.P. Robinson, G. Rosolini: The discrete objects in the effective topos, *Proc. Lond. Math. Soc.* (3) 60 (1990) 1–36.

[15] T. Jim, A. R. Meyer: Full abstraction and the context lemma (preliminary report), in: *Theoretical Aspects of Computer Software (Sendai, September 1991)* (eds. T. Ito, A. R. Meyer), Springer LNCS 526, 1991.

[16] P. T. Johnstone: *Topos Theory*, Academic Press, 1977.

[17] B. Lercher: *Strong Reduction and Recursion in Combinatory Logic*, dissertation, Pennsylvania State University, 1963.

[18] G. Longo, E. Moggi: Constructive natural deduction and its "ω-set" interpretation, LIENS–90–21, October 1990; to appear in *Math. Structures in Comp. Sci.*

[19] J. C. Mitchell: On abstraction and the expressive power of programming languages, in *Proc. of TACS '91 (Sendai)*, Springer LNCS 526.

[20] W. K.-S. Phoa: Effective domains and intrinsic structure, in: Proc. of 5th Annual Symposium on Logic in Computer Science, 1990.

[21] W. K.-S. Phoa: Building domains from graph models, to appear in *Math. Structures in Comp. Sci.*

[22] W. K.-S. Phoa: *Domain Theory in Realizability Toposes*, dissertation, University of Cambridge, November 1990; available as CST-82-91/ECS-LFCS-91-171, Department of Computer Science, University of Edinburgh.

[23] W. K.-S. Phoa: Two results on set-theoretic polymorphism, in *Proc. of Category Theory and Computer Science (Paris, 1991)*, Springer LNCS.

[24] W. K.-S. Phoa: PCF and the untyped λ-calculus, submitted.

[25] W. K.-S. Phoa: Using fibrations to understand subtypes, to appear in *Proc. of the Durham Symposium on Applications of Category Theory to Computer Science*, London Mathematical Society.

[26] W. K.-S. Phoa, P. Taylor: The synthetic Plotkin powerdomain, preprint.

[27] G. D. Plotkin: LCF considered as a programming language, *Theor. Comp. Sci.* 5(3):223–256.

[28] G. D. Plotkin, M. B. Smyth: The category-theoretic solution of recursive domain equations, *Siam Journal of Computing* vol. 11 no. 4, 1982.

[29] E. Robinson: How complete is PER? in: Proc. of 4th Annual Symposium on Logic in Computer Science, 1989.

[30] G. Rosolini: Categories and effective computations, in: Lecture Notes in Computer Science 283, Springer, 1987.

[31] D.S. Scott: Data types as lattices, *SIAM J. Comput.* 5 (3) (1976) 522-587.