Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory

Citation for published version:
Simpson, A 2002, Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory. in *Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on .* Institute of Electrical and Electronics Engineers (IEEE), pp. 287-298. https://doi.org/10.1109/LICS.2002.1029837

Digital Object Identifier (DOI):
10.1109/LICS.2002.1029837

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on

Simpson, A. (2002). Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory. In Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on . (pp. 287-298). IEEE Computer Society. doi: 10.1109/LICS.2002.1029837

Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory

Alex Simpson*
LFCS, Division of Informatics,
University of Edinburgh, Scotland

Abstract

We present a general axiomatic construction of models of FPC, a recursively typed lambda-calculus with call-by-value operational semantics. Our method of construction is to obtain such models as full subcategories of categorical models of intuitionistic set theory. This allows us to obtain a notion of model that encompasses both domain-theoretic and realizability models. We show that the existence of solutions to recursive domain equations, needed for the interpretation of recursive types, depends on the strength of the set theory. The internal set theory of an elementary topos is not strong enough to guarantee their existence. However, solutions to recursive domain equations do exist if models of intuitionistic Zermelo-Fraenkel set theory are used instead, We apply this result to interpret FPC, and we provide necessary and sufficient conditions on a model for the interpretation to be computationally adequate, i.e. for the operational and denotational notions of termination to agree.

1. Introduction

In this paper, we present a general axiomatic account of the construction of denotational models of FPC, a recursively-typed λ-calculus with sum and product types. A vital property of a model is that it should be *computationally adequate*, i.e. that the denotational account of termination should coincide with the operational one. We provide necessary and sufficient conditions on a model for computational adequacy to hold.

Because FPC is a typed functional language, its models are necessarily categories. In fact, one can identify exactly the structure required by a category, $\mathbf{P}$, to model the language. It must have: finite sums and products, to interpret the corresponding type constructors; a lifting monad, $\mathbb{L}$, to account for the possible nontermination of programs; partial (relative to $\mathbb{L}$) exponentials, to interpret function types; and finally, to interpret recursive types, the derived category, $\mathbf{pP}$, of partial maps, induced by $\mathbb{L}$ on $\mathbf{P}$, must be *algebraically compact* in the sense of Freyd [7, 8], at least with repect to functors defined by type expressions.

The above identifies the structure required by a model of FPC, but does not indicate where to find examples of models. Nevertheless, several sources of such models are known. Domain theory provides the classical example of the category of ωcpos [24]. More generally, axiomatic domain theory has successfully abstracted the idiosyncracies of domains to provide a host of "neo-classical" models [2, 4]. A quite different type of model is given by game-theoretic semantics [18]. Finally, while the structure has not previously been exhibited in the form above, it has long been known that there should be a variety of models based on realizability semantics [9, 20, 21, 22, 17]. What has been missing hitherto is a single unifying treatment accounting for the existence of all these types of model. In this paper, we provide such a treatment.

In [28], Dana Scott observed that categories of domains can live as full subcategories of models of intuitionistic set theory. We exploit this idea to construct models of FPC in a uniform way. Roughly speaking, we start off with a category $\mathbf{S}$ of intuitionistic sets that satisfies one simple axiom, Axiom $\mathbf{N}$ of Section 2, which, although classically inconsistent, is intuitionistically consistent. From any such category $\mathbf{S}$, we extract a full subcategory of *predomains*, $\mathbf{P} \hookrightarrow \mathbf{S}$, with all the structure identified above, and hence we have a model of FPC.

This approach directly follows [30], where it is shown that a model of the simply-typed language PCF [23] can be similarly extracted from any elementary topos $\mathbf{S}$ (with natural numbers object) satisfying Axiom $\mathbf{N}$. The additional goal of the present paper is to show that $\mathbf{P}$ also models recursive types. This is a non-trivial task.

In fact, we immediately encounter a problem. As our first result, Proposition 1, we show that there exists an elementary topos satisfying Axiom $\mathbf{N}$ for which the derived category $\mathbf{pP}$ is *not* algebraically compact. Thus some mod-

*Research supported by EPSRC Research Grant no. K06109 and an EPSRC Advanced Research Fellowship.

ification to the above method of constructing $\mathbf{P}$ is necessary in order to interpret recursive types. This is not, at first sight, surprising. Axiom $\mathbf{N}$ is designed merely to guarantee that $\mathbf{P}$ models the recursive definition of functions. Thus there is no *a priori* reason to expect recursive types to have interpretations in $\mathbf{pP}$.

However, we identify the difficulty as stemming from a perhaps unexpected source. The problem is that elementary toposes, although models of intuitionistic higher-order logic, are not, in general, models of a sufficiently powerful set theory. Thus, instead of working with an arbitrary elementary topos, we shall require that $\mathbf{S}$ have enough structure to model full *Intuitionistic Zermelo-Fraenkel* (IZF) set theory, see e.g. [27]. Technically, this is implemented by asking for $\mathbf{S}$ to be given as the full subcategory of *small* objects in a category $\mathbf{C}$ with *class(ic) structure* and *universal object*, in the sense of [31] (developed from [15]). As our first main result, Theorem 1, we prove that, with such a category $\mathbf{S}$, the derived category $\mathbf{pP}$ is algebraically compact whenever Axiom $\mathbf{N}$ holds. Thus, with enough set-theoretic power to back it up, Axiom $\mathbf{N}$ is, after all, sufficient for the solution of recursive domain equations.

The proof of Theorem 1 occupies Sections 4–6. An informal outline of the proof structure, including a discussion of the technical innovations required, is given in Section 3.

By Theorem 1, it is possible to interpret FPC in $\mathbf{P}$. We give the interpretation explicitly in Sections 7 and 9. Recall that the interpretation is said to be *computationally adequate* if the denotational account of program termination coincides with actual termination in the operational semantics. As Theorem 2, we prove that the interpretation of FPC is computationally adequate if and only if the internal logic of $\mathbf{S}$ is 1-consistent (i.e. only genuinely true Σ^0_1-sentences are true in $\mathbf{S}$). Thus the programming-language-sensitive property of computational adequacy is reduced to a purely logical property of $\mathbf{S}$. This result is based on the similar characterisation of computational adequacy for PCF in [30]. However, the extension of the result to FPC is non-trivial, see Section 8.

Finally, in Section 10, we present applications of our work across the range models discussed earlier. The classical domain-theoretic models, such as the category of ωcpos, and their generalizations [2, 4], all embed in Grothendieck toposes [3, 5], and hence, by [15, Ch. IV], in categories with class structure. Moreover, under mild conditions, Axiom $\mathbf{N}$ is satisfied. Also, by their very definition, realizability models [9, 20, 21, 22, 16, 17] embed in realizability toposes [10, 12], and hence in categories with class structure [15, Ch. IV]. Again, Axiom $\mathbf{N}$ is satisfied. Thus, Theorem 1 gives an account of the construction of solutions to recursive domain equations that applies simultaneously to domain-theoretic and to realizability models.

As all nontrivial Grothendieck and realizability toposes are 1-consistent, we obtain a uniform proof of computational adequacy for the models discussed above. For domain-theoretic models, computational adequacy has previously only been proved in an order-enriched setting [2], whereas our result applies also to the more general class of enriched models axiomatized in [4, 3]. For realizability models, the only existing proof of computational adequacy for a language (implicitly) containing recursive types, applies to just one specific model [1]. We thus obtain the first proof of computational adequacy, for the interpretation of a language with recursive types, in all the realizability models of [9, 20, 21, 22, 16, 17].

Acknowledgements This paper was conceived during a visit to Genova in April 1995, for which I express my warm thanks to Pino Rosolini. Over the lengthy period of its development, I have benefited from discussions with Marcelo Fiore, Edmund Robinson, Pino Rosolini, Thomas Streicher and Paul Taylor, the last of whom is also acknowledged for providing the macros used to format the diagrams.

2. Classes, sets and predomains

As discussed in the introduction, our work will involve both elementary toposes and also categorical models of Intuitionistic Zermelo-Fraenkel (IZF) set theory [27]. Both types of model arise as instances of regular categories with *class(ic) structure*, as defined in [31]. We briefly recount the main features of this notion, using, as far as posible, set-theoretic intuition. For the category-theoretic details see *op. cit.*

In a regular category,[1] $\mathbf{C}$, with class structure, the objects are to be thought of as classes and the morphisms as functions between classes. There is a distinguished full subcategory, $\mathbf{S}$, of *small* objects, which is to be thought of as the subcategory of sets. More generally, there is a distinguished collection of morphisms, the *small maps*, where intuitively $f\colon X \longrightarrow Y$ is small if, for every y in the class Y, its fibre $f^{-1}(y)$, which is a subclass of the class X, is actually a set. Smallness interacts with the regular structure on $\mathbf{C}$ as follows. If $X \rightarrowtail Y$ is mono and Y is small then X is small, i.e. every subclass of a set is a set. This expresses the Separation axiom of set theory. Dually, if $X \twoheadrightarrow Y$ is epi (n.b. class structure implies that every epi is regular) and X is small then Y is small, i.e. the image of a function from a set to a class is itself a set. This expresses the Replacement axiom of set theory. The other important structure on $\mathbf{C}$ is that, for every class X, there is another class $\mathcal{P}_{\mathbf{S}}X$ the *small powerobject* of X, which is intuitively the class of all subsets of X. The object $\mathcal{P}_{\mathbf{S}}X$ comes with an associated *membership* relation $\ni_X \rightarrowtail \mathcal{P}_{\mathbf{S}}X \times X$, for which the

[1] A *regular category* is a category with finite limits in which every morphism has a stable factorization as a regular epi followed by a mono.

composite

$$\gamma_X \quad = \quad \exists_X \rightarrowtail \mathcal{P}_{\mathbf{S}} X \times X \xrightarrow{\pi_1} \mathcal{P}_{\mathbf{S}} X \qquad (1)$$

is a small map. It is also required that if X is small then so is $\mathcal{P}_{\mathbf{S}} X$. This expresses the Powerset axiom of set theory. It follows that the full subcategory $\mathbf{S}$ is an elementary topos. Further, $\mathbf{C}$ has finite coproducts and $\mathbf{S}$ is closed under finite limits and coproducts in $\mathbf{C}$. We shall make liberal use of the internal logic of $\mathbf{C}$, which is intuitionistic first-order logic, with the predicates on X being arbitrary subobjects of X. We write $\mathbf{C} \models \varphi$ to mean that statement φ holds in the internal logic of $\mathbf{C}$. The object $\Omega = \mathcal{P}_{\mathbf{S}}\mathbf{1}$ (where $\mathbf{1}$ is the terminal object in $\mathbf{C}$), which is the subobject classifier in $\mathbf{S}$, is also a subobject classifier in $\mathbf{C}$. Thus Ω can be thought of as the set of all internal propositions in $\mathbf{C}$.

As we shall make heavy use of indexed families in $\mathbf{C}$, we summarise the legitimate constructions on them in the context of class structure. As usual, we consider I-indexed families as being given by morphisms $X \longrightarrow I$, although we shall often use the convenient notation $\{X_i\}_{i:I}$ for them. Given such an internal family $X \longrightarrow I$, the object X itself provides a dependent sum $\sum_{i:I} X_i$. However, a dependent product $\prod_{i:I} X_i$ is only guaranteed to exist in the case that I is a small object. If, in addition to I being small, $X \longrightarrow I$ is a small map then $\prod_{i:I} X_i$ is itself a small object. In the case of a constant families $\{X\}_{y:Y}$ (given by projections $X \times Y \longrightarrow Y$), dependent products specialise to function spaces. Thus the above remarks imply that Y^X exists whenever X is a small object, and that Y^X is itself small if both X and Y are small.

Henceforth in this paper, let $\mathbf{C}$ be a regular category with class structure, and let $\mathbf{S}$ be its full subcategory of small objects. Further, we assume that $\mathbf{C}$ has a small natural numbers object (nno) $\mathbf{N}$. This implements the Infinity axiom of set theory. However, in spite of the motivating references to set theory, the assumed structure on $\mathbf{C}$ and $\mathbf{S}$ does not yet provide the full power of IZF set theory. For example, given any elementary topos with nno, $\mathbf{S}$, one can obtain class structure by putting $\mathbf{C} = \mathbf{S}$ and stipulating that every map be small.

The remaining goal of this section is to isolate a full subcategory of $\mathbf{S}$ to act as a category of predomains. This will require imposing further axioms on $\mathbf{C}$. Many axiomatizations have been proposed for this purpose, see e.g. [26, 11, 20, 33, 17, 30, 25, 19]. Here, we follow [30].

As first proposed in [26], the definition of predomain is predicated on a notion of partiality. To implement this, we require a distinguished subobject $\Sigma \rightarrowtail \Omega$. Intuitively Σ corresponds to the subobject of those propositions in Ω that express the termination of programs. As Σ is a subobject of Ω, it classifies a collection of subobjects in $\mathbf{C}$, namely those whose characteristic map to Ω factors through $\Sigma \rightarrowtail \Omega$. Intuitively, such Σ-*subobjects* of X correspond to those subobjects determined as the domains of termination of programs taking input in X. Because there exist terminating programs, and because programs can be run under sequential composition, it makes sense to require that Σ contains the true proposition, $\top$, and that Σ-subobjects are closed under composition. This implies, in particular, that Σ is closed under finite conjunction in Ω. Taken together, these requirements state that Σ is a *dominance* [26].

The dominance Σ determines a *lifting* functor on $\mathbf{C}$. For an object X, we say $e : \mathcal{P}_{\mathbf{S}} X$ is *subterminal* if

$$\forall x, x' : X.\ x \in e \wedge x' \in e \rightarrow x = x'.$$

We say that e is Σ-*subterminal* if it is subterminal and also

$$(\exists x : X.\ x \in e) \in \Sigma,$$

i.e. the proposition stating that e is inhabited is a Σ-proposition. Using the internal logic of $\mathbf{C}$, define

$$\mathbb{L}X = \{e : \mathcal{P}_{\mathbf{S}} X \mid e \text{ is } \Sigma\text{-subterminal}\}.$$

The $\mathbb{L}$ operation extends to a functor $\mathbb{L} : \mathbf{C} \to \mathbf{C}$, where, on $f : X \longrightarrow Y$, the morphism action $\mathbb{L}f : \mathbb{L}X \longrightarrow \mathbb{L}Y$ is defined by

$$(\mathbb{L}f)(e) = \{f(x) \mid x \in e\}.$$

Further, the endofuctor $\mathbb{L}$ carries a monad structure. The unit is singleton $\{\cdot\} : X \longrightarrow \mathbb{L}X$, and the multiplication is union $\bigcup : \mathbb{L}\mathbb{L}X \longrightarrow \mathbb{L}X$.

As in [14], the endofunctor $\mathbb{L}$ has a final coalgebra, $\tau : \mathbf{F} \longrightarrow \mathbb{L}\mathbf{F}$ (necessarily an isomorphism), defined by:

$$\begin{aligned} \mathbf{F} &= \{c : \Sigma^{\mathbf{N}} \mid \forall n : \mathbf{N}.\ c(n+1) \to c(n)\} \\ \tau(c) &= \{(n \mapsto c(n+1)) \mid c(0)\}. \end{aligned}$$

Because $\mathbf{F}$ is small, there exists a smallest subalgebra, $\sigma : \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$, of τ^{-1}, defined internally in $\mathbf{C}$ as the intersection of all subalgebras of τ^{-1}. It is a consequence of [31, Theorem 5] that $\sigma : \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$ is an initial algebra for the endofunctor $\mathbb{L}$ on $\mathbf{C}$. By construction, the unique algebra homomorphism, $\iota : \mathbf{I} \longrightarrow \mathbf{F}$, from σ to τ^{-1} is mono.

One can view $\mathbf{I}$ as the object obtained from the initial object $\mathbf{0}$ by freely iterating the $\mathbb{L}$ functor. In the sequel, $\mathbf{I}$ will play the rôle of a generic "ω-chain" in $\mathbf{C}$, and $\mathbf{I} \rightarrowtail \mathbf{F}$ will exhibits $\mathbf{F}$, which has the additional "infinite" point $\infty = (n \mapsto \top)$, as its "chain-completion". This intuition plays a fundamental rôle in developing a basic notion of "chain completeness" used to define a full subcategory of predomains within $\mathbf{S}$, see [17].

Definition 2.1 (Complete object) An object X is *complete* if $X^\iota : X^{\mathbf{F}} \longrightarrow X^{\mathbf{I}}$ is an isomorphism.

Examples in [19] show that complete objects do not themselves form a suitable category of predomains as they are not necessarily closed under lifting. Following [17], we avoid this problem usng the property of *well-completeness*.

Definition 2.2 (Well-complete object) An object X is *well-complete* if $\mathbb{L}X$ is complete.

The results below, which are standard, see e.g. [30], state the basic properties of well-completeness. In them, we write $\mathbf{2}$ for the object $\mathbf{1}+\mathbf{1}$, which we view as a subobject of Ω via $[\bot,\top]\colon \mathbf{2} \rightarrowtail \Omega$, where $\bot$ is *falsum*.

Lemma 2.3

1. *If $\mathbf{2}$ is well-complete then so are $\mathbf{1}$ and $\mathbf{0}$.*
2. *If $\mathbf{N}$ is well-complete then so is $\mathbf{2}$.*

The converse implications do not hold in general, see [19].

Lemma 2.4 *If $\mathbf{1}$ is well-complete then:*

1. *X well-complete implies X complete.*
2. *X well-complete implies $\mathbb{L}X$ well-complete.*
3. *For any internal family $\{X_i\}_{i:\,I}$ with I small,*

$$\mathbf{C} \models (\forall i\!:\!I.\ X_i \text{ is well-complete}) \rightarrow (\prod_{i:I} X_i) \text{ is well-complete.}$$

   Two special cases:

   If X, Y are well-complete then so is $X \times Y$.

   If X is small and Y is well-complete then Y^X is well-complete.

4. *Given two morphisms $f,g\colon X \longrightarrow Y$ with X,Y well-complete then, in the equaliser $e\colon E \rightarrowtail X$ of f and g, the object E is well-complete.*
5. *$\mathbf{0}$ is well-complete if and only if $\bot \in \Sigma$.*
6. *$\mathbf{2}$ is well-complete if and only if X,Y well-complete implies $X+Y$ well-complete.*
7. *$\mathbf{N}$ is well-complete if and only if $\mathbf{2}$ is well-complete and also*

$$\mathbf{C} \models \forall P : \mathbf{2}^{\mathbf{N}}.\ (\exists n\!:\!\mathbf{N}.\ P(n)) \in \Sigma\,. \qquad (2)$$

Here, statement 3 makes use of the fact that well-completeness can be formulated in the internal logic. Also (2) states that, for any logically decidable predicate P on $\mathbf{N}$, the proposition $\exists n\!:\!\mathbf{N}.\ P(n)$ is a Σ-proposition.

In this paper, a *predomain* is simply a small well-complete object. We write $\mathbf{P}$ for the full subcategory of predomains. Thus we have full subcategory inclusions $\mathbf{P} \hookrightarrow \mathbf{S} \hookrightarrow \mathbf{C}$. For $\mathbf{P}$ to be well behaved, we need axioms to assume that basic objects are predomains. As all the obects we consider for this purpose are already small, the axioms are formulated in terms of well-completeness alone. We use a single format for all axioms.

Axiom X The object $\mathbf{X}$ is well-complete.

We shall instantiate this format in three instances only: Axiom $\mathbf{1}$, which, by Lemma 2.4.3, implies that $\mathbf{P}$ is cartesian closed; Axiom $\mathbf{2}$ which, by Lemma 2.4.6, implies that, $\mathbf{P}$ has finite coproducts (inherited from $\mathbf{C}$); and Axiom $\mathbf{N}$, which, as is shown in [30], implies that $\mathbf{P}$ has all the structure required by a model of PCF. The implications between these three axioms are given by Lemma 2.3.

Our goal, in this paper, is to address the interpretation of recursive types in $\mathbf{P}$. This requires that recursive domain equations have solutions up to isomorphism in an associated category $\mathbf{pP}$ of partial maps, which we now define.

For objects X, Y of $\mathbf{C}$, a Σ*-partial map* is a partial map from X to Y whose domain $X' \rightarrowtail X$ is a Σ-subobject of X. Because Σ is a dominance, Σ-partial maps are closed under composition. As the only partial maps we are interested in are Σ-partial, we henceforth drop the Σ. We write $\mathbf{pC}$ for the category of partial maps between objects of $\mathbf{C}$, and we write $\mathbf{pP}$ for the full subcateory of $\mathbf{pC}$ on predomains. We write $X \rightharpoonup Y$ for the object of partial maps from X to Y, which is easily defined in the internal logic. The object $X \rightharpoonup Y$ is isomorphic to the exponential $(\mathbb{L}Y)^X$. Thus, by Lemma 2.4, if Axiom $\mathbf{1}$ holds then, for X small and Y a predomain, $X \rightharpoonup Y$ is a predomain.

The first new result of this paper shows that, in the context of the assumed structure on $\mathbf{C}$, Axiom $\mathbf{N}$ is not sufficient to allow recursive domain equations to be solved in $\mathbf{pP}$. The statement makes use of the fact, already discussed, that any elementary topos $\mathbf{S}$ arises as the full subcategory of small objects in a category with class structure, by taking $\mathbf{C} = \mathbf{S}$.

Proposition 1 *There is an elementary topos satisfying Axiom $\mathbf{N}$ in which there exists a predomain Υ such that no solution X to the isomorphism $X \cong X \rightharpoonup \Upsilon$ exists in $\mathbf{pP}$.*

We just state what the example is. Let ϖ be the set of ordinals $\leq \omega$, with their usual ordering, endowed with the Scott topology. The Grothendieck topos $\mathcal{H}$, from [5], is the topos of sheaves over the canonical Grothendieck topology on the monoid of continuous endofunctions on ϖ. Let $\mathcal{H}_{\beth_\omega}$ be the full subcategory of $\mathcal{H}$ on those sheaves A for which the set $A(\varpi)$ has cardinality strictly less than $\beth_\omega$, where $\beth_\omega = \sup\{2^{\aleph_0}, 2^{2^{\aleph_0}}, 2^{2^{2^{\aleph_0}}}, \ldots\}$. As in [5], there is a full embedding $\mathbf{y} : \omega\mathbf{cpo}_{\beth_\omega} \hookrightarrow \mathcal{H}_{\beth_\omega}$ of the category of ω-cpos of cardinality $< \beth_\omega$ in $\mathcal{H}_{\beth_\omega}$. Using this, define $\Sigma = \mathbf{y}(\mathbb{O})$, where $\mathbb{O}$ is Sierpinski space. Then, as in [5], Axiom $\mathbf{N}$ is satisfied. Finally, define $\Upsilon = \mathbf{y}(Z)$ where Z is the ωcpo (the well-known countably-based L-domain that is not bifinite) drawn in [34, Example 9.6.15(c)]. One can show that any solution X to $X \cong X \rightharpoonup \Upsilon$ would have $|X(\overline{\omega})| \geq \beth_\omega$, hence no such solution exists in $\mathcal{H}_{\beth_\omega}$.

3. Algebraic compactness

As indicated in the introduction, we address the interpretation of recursive types by strengthening the assumptions on our ambient category of classes $\mathbf{C}$. A *universal object* is an object U such that, for every object X, there exists a mono $X \rightarrowtail U$. Thus U can be thought of as an object that collects the elements of all classes together within one universal class. In set-theoretic terms, U is simply the class of all sets (and atoms if permitted). In [31] it is shown how the existence of a universal object implies that $\mathbf{C}$ contains an internal model of IZF set theory.

Henceforth we require that $\mathbf{C}$ have a universal object. For the purposes of this paper, a vital consequence of the universal object is that the categories $\mathbf{S}$, $\mathbf{P}$ and $\mathbf{pP}$ all live as internal categories within $\mathbf{C}$.

As usual, an internal category, K, in $\mathbf{C}$ is given by an object (i.e. a class), $|\mathsf{K}|$, of K-objects, and an internal family, $\{\mathsf{K}(A,B)\}_{A,B\,:\,|\mathsf{K}|}$, of K-morphisms indexed by domain and codomain, satisfying the expected axioms for identities and composition, see e.g. [13]. We say that an internal category K in $\mathbf{C}$ is *locally small* if the internal family

$$\{\mathsf{K}(A,B)\}_{A,B\,:\,|\mathsf{K}|} \longrightarrow |\mathsf{K}| \times |\mathsf{K}|$$

is a small map in $\mathbf{C}$. It is *small* if, in addition, $|\mathsf{K}|$ is small.

An *internal functor*, F, from an internal category K to another L is given by a morphism

$$F \colon |\mathsf{K}| \longrightarrow |\mathsf{L}|,$$

expressing the action on objects, together with a family

$$\{F_{A,B} \colon \mathsf{K}(A,B) \longrightarrow \mathsf{L}(FA,FB)\}_{A,B\,:\,|\mathsf{K}|}$$

that preserves identities and composition, again see [13].

We briefly exhibit $\mathbf{S}$ as an internal category in $\mathbf{C}$, before turning attention to $\mathbf{P}$ and $\mathbf{pP}$, which are the categories of interest to us. The internal category S is defined by

$$|\mathsf{S}| \;=\; \mathcal{P}_{\mathbf{S}}U \qquad\qquad \mathsf{S}(A,B) \;=\; B^A,$$

where the family $\{B^A\}_{A,B\,:\,\mathcal{P}_{\mathbf{S}}U}$ is defined as an exponential of small objects in the slice category $\mathbf{C}/(\mathcal{P}_{\mathbf{S}}U \times \mathcal{P}_{\mathbf{S}}U)$. Identities and composition are defined in the obvious way. By the earlier remarks on smallness and function spaces, S is a locally small internal category in $\mathbf{C}$. Using the theory of fibrations, see [13], one can formulate a precise statement that the category $\mathbf{S}$ is the *externalization* of the internal category S.

Analogously, we next construe both $\mathbf{P}$ and $\mathbf{pP}$ as internal categories P and pP respectively. First we define P by

$$|\mathsf{P}| = \{A \colon \mathcal{P}_{\mathbf{S}}\, U \mid A \textit{ is a predomain}\} \quad \mathsf{P}(A,B) = B^A\,,$$

using the evident formulation of the property of being a predomain in the internal logic of $\mathbf{C}$. Thus P is an internal full subcategory of S, and hence locally small. The internal category pP is defined by

$$|\mathsf{pP}| \;=\; |\mathsf{P}| \qquad\quad \mathsf{pP}(A,B) \;=\; A \rightharpoonup B\,,$$

with the obvious identities and composition. Again, pP is locally small (because $A \rightharpoonup B \cong (\mathbb{L}B)^A$).

As before, using the theory of fibrations, one can make precise that $\mathbf{P}$ and $\mathbf{pP}$ are the externalizations of P and pP respectively. A crucial consequence of such externalization results is that fibred (over $\mathbf{C}$) structure on $\mathbf{P}$ and $\mathbf{pP}$ gives rise to corresponding internal structure on the internal categories P and pP. For example, assuming Axiom $\mathbf{1}$, the $\mathbb{L}$-monad on $\mathbf{P}$ determines an internal monad $(L, \{\cdot\}, \bigcup)$ on P. Also, Lemma 2.4 can be interpreted as an internal proposition about the internal category pP. Statements 3 and 4 of the proposition together imply that, in the presence of Axiom $\mathbf{1}$, it holds in $\mathbf{C}$ that the internal category P is *small-complete*[2], with limits inherited from S. Thus there are morphisms in $\mathbf{C}$ that find limiting cones for small diagrams in P. The internal category pP is not small-complete. Nevertheless, one can derive internal functors:

$$\mathsf{pP} \times \mathsf{pP} \xrightarrow{\;\times\;} \mathsf{pP} \tag{3}$$

$$\mathsf{pP}^{op} \times \mathsf{pP} \xrightarrow{\;\rightharpoonup\;} \mathsf{pP} \tag{4}$$

$$\mathsf{pP} \times \mathsf{pP} \xrightarrow{\;+\;} \mathsf{pP}\,, \tag{5}$$

where (3) and (4) require Axiom $\mathbf{1}$, and (5) requires Axiom $\mathbf{2}$. N.b. although $\times$ extends product on P, it is *not* a cartesian product on pP, whereas $+$ is a binary coproduct functor on pP.

Our goal is to prove the algebraic compactness, in the sense of Freyd [7, 8], of the internal category pP. We recall this notion for ordinary categories. Given an endofunctor F on an arbitrary category $\mathcal{K}$, a *bifree* algebra is an initial F-algebra $a \colon FA \longrightarrow A$ for which a^{-1} is also a final F-coalgebra (by Lambek's Lemma, an initial algebra is always an isomorphism). A category $\mathcal{K}$ is said to be *algebraically compact* if every endofunctor on it has a bifree algebra.

The correct formulation of algebraic compactness for an internal category K in $\mathbf{C}$ is slightly subtle because there

[2] N.b. P, although locally small, is not a small internal category.

need not be any object of all K-endofunctors in $\mathbf{C}$ to allow an internal universal quantification. Instead, we make an external quantification over internal families of internal functors. Technically, this ensures that the definition is stable under the formation of slice categories of $\mathbf{C}$.

Definition 3.1 (Algebraic compactness) An internal category K is said to be *algebraically compact* if, for every internal family $\{F_i \colon \mathsf{K} \to \mathsf{K}\}_{i\,:\,I}$ in $\mathbf{C}$ of internal endofunctors, there exists a morphism $A_{(-)} \colon I \longrightarrow |\mathsf{K}|$, and a family $\{a_i \colon \mathsf{K}(F_i A_i, A_i)\}_{i\,:\,I}$ such that

$$\mathbf{C} \models \forall i\colon I.\ a_i \text{ is a bifree } F_i\text{-algebra.}$$

Moreover, the above data must be preserved by reindexing: i.e., for $f \colon J \longrightarrow I$ in $\mathbf{C}$, let $B_{(-)} \colon J \longrightarrow |\mathsf{K}|$ and $\{b_j \colon \mathsf{K}(F_i B_j, B_j)\}_{j\,:\,J}$ be determined, as above, by the J-indexed family $\{F_{f(j)} \colon \mathsf{K} \to \mathsf{K}\}_{j\,:\,J}$, then it must hold that $B_{(-)} = A_{(-)} \circ f$ and $b_{(-)} = a_{(-)} \circ f$.

Lemma 3.2 (Parametrized algebraic compactness) *Suppose K and L are internal categories with K algebraically compact, and suppose $F \colon \mathsf{L} \times \mathsf{K} \to \mathsf{K}$ is an internal functor. Let $A_{(-)} \colon |\mathsf{L}| \longrightarrow |\mathsf{K}|$ and $\{a_B \colon \mathsf{K}(F(B, A_B), A_B)\}_{B\,:\,|\mathsf{L}|}$ be the data given by algebraic compactness, viewing F as indexed over $|\mathsf{L}|$. Then there exists a unique internal functor $F^\dagger \colon \mathsf{L} \to \mathsf{K}$ such that $F^\dagger B = A_B$ and $a_B \colon F(B, F^\dagger B) \cong F^\dagger B$ is natural in B.*

Theorem 1 *If Axiom* **1** *holds then the internal category* pP *is algebraically compact.*

The proof of Theorem 1 occupies Sections 4–6. The strategy is to establish a version of the limit-colimit coincidence of classical domain theory (see, e.g. [32]), and apply it to pP. However, a major complication arises. In many realizability models of our setting, the usual limit-colimit-coincidence is simply false, at least when formulated using diagrams indexed by the natural numbers $\mathbf{N}$, see [19] for a counterexample. We solve this problem by developing a non-trivial variant, under which diagrams are indexed by the carrier $\mathbf{I}$ of the initial-algebra structure for $\mathbb{L}$. This is presented as Proposition 2.

We apply Proposition 2, by developing sufficient conditions for an internal category K in $\mathbf{C}$ to be algebraically compact, Proposition 3. A first crucial feature here is that $|\mathsf{K}|$, the class of objects of K, should carry an algebra structure for the $\mathbb{L}$ functor. This allows $\mathbf{I}$-indexed diagrams, of the form required by the limit-colimit coincidence, to be constructed using the initial algebra property of $\mathbf{I}$. A second crucial feature is that all such diagrams must have a limit in pP, or equivalently a colimit.

Finally, as Proposition 4, we show that pP does indeed satisfy the conditions of Proposition 3. This concludes the proof of Theorem 1.

4. Pointed objects and multistrict maps

As crucial preparation for the proof of Theorem 1, we use the lifting monad to implement a notion of pointed object, and of strict map between pointed objects. For us, a *pointed object* (X, α) is simply an Eilenberg-Moore algebra $\alpha \colon \mathbb{L}X \longrightarrow X$ for the monad $(\mathbb{L}, \{\cdot\}, \bigcup)$. If $\bot \in \Sigma$ then one can think of $\alpha(\emptyset)$ as the identified "point" of X, but the notion of pointed object also makes sense without the assumption that $\bot \in \Sigma$. A *strict map* $h \colon (X, \alpha) \longrightarrow (Y, \beta)$ between pointed objects is simply an algebra homorphism (i.e. a morphism $h \colon X \longrightarrow Y$ such that $h \circ \alpha = \beta \circ \mathbb{L}h$). Given pointed objects $(X_1, \alpha_1), \ldots, (X_k, \alpha_k)$ and (Y, β), a *k-strict map* is a morphism $h \colon X_1 \times \cdots \times X_k \longrightarrow Y$ such that, for each i with $1 \le i \le k$, it holds in $\mathbf{C}$ that

$$\forall x_1 \colon X_1, \ldots, x_{i-1} \colon X_{i-1},\ x_{i+1} \colon X_{i+1}, \ldots, x_k \colon X_k.\\ x_i \mapsto h(x_1, \ldots, x_k) \text{ is a strict map from } X_i \text{ to } Y.$$

We use *bistrict* for the cases $k = 2$, and *multistrict* if we leave k implicit. The lemma below is a special feature of lifting monads.

Lemma 4.1 *Given pointed objects (X_1, α_1), (X_2, α_2) and (Y, β), then any bistrict map $h \colon X_1 \times X_2 \longrightarrow Y$ is a strict map from the pointed object $X_1 \times X_2$ to Y.*

The initial algebra of the endofunctor $\mathbb{L}$ caries a pointed structure $\phi = \sigma \circ \bigcup \circ \mathbb{L}\sigma^{-1} \colon \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$. The pointed structure on $\mathbf{I}$ interacts nicely with the initial algebra property. Define a "successor" function $s = \sigma \circ \{\cdot\} \colon \mathbf{I} \longrightarrow \mathbf{I}$. The lemma below generalizes [15, Theorem A.5].

Lemma 4.2 *Suppose that (X, α) is a pointed object and that $f : X \longrightarrow X$ is any (not necessarily strict) morphism. Then, for every $k \ge 1$, there exists a unique k-strict map $h : \mathbf{I}^k \longrightarrow X$ such that the diagram below commutes.*

$$\begin{tikzcd} \overbrace{\mathbf{I} \times \cdots \times \mathbf{I}}^{k} \arrow[r, "h"] \arrow[d, "s \times \cdots \times s"'] & X \arrow[d, "f"] \\ \mathbf{I} \times \cdots \times \mathbf{I} \arrow[r, "h"] & X \end{tikzcd}$$

Using Lemma 4.2, define $min : \mathbf{I} \times \mathbf{I} \longrightarrow \mathbf{I}$ to be the unique bistrict map such that $min(si, sj) = s(min(i, j))$. Then, by Lemmas 4.1 and Lemma 4.2 in the cases $k = 1, 3$,

$$min(i, i) = i$$
$$min(i, min(j, k)) = min(min(i, j), k).$$

Thus min gives an internal semilattice structure to $\mathbf{I}$. In the standard way, we use this to define an internal partial order on $\mathbf{I}$ by

$$i \sqsubseteq j \text{ iff } i = min(i,j).$$

The next lemma, which play an important rôle in the sequel, seems very much a peculiarity of lifting monads.

Lemma 4.3 *Given an internal family $\{(Y_x, \beta_x)\}_{x\,:\,X}$ of pointed objects, where (X, α) is also pointed, then so is $(\sum_{x\,:\,X} Y_x, \gamma)$, where $\gamma\colon \mathbb{L}(\sum_{x\,:\,X} Y_x) \longrightarrow \sum_{x\,:\,X} Y_x$ is defined by $\gamma(e) = (\gamma_1(e), \gamma_2(e))$, where*

$$\gamma_1(e) = \alpha\{x \mid (x,y) \in e\}$$
$$\gamma_2(e) = \beta_{\gamma_1(e)}\{y \mid (x,y) \in e\}.$$

We shall also need a notion of strictness for dependent families, which again seems peculiar to lifting monads.

Definition 4.4 (Strict family) Given an internal family $\{(Y_x, \beta_x)\}_{x\,:\,X}$ of pointed objects, where (X, α) is also pointed, we say that $y_{(-)}\colon \prod_{x\,:\,X} Y_x$ is a *strict family* if, for all $e\colon \mathbb{L}X$,

$$y_{\alpha(e)} = \beta_{\alpha(e)}\{y_x \mid x \in e\}.$$

The above definition relates to Lemma 4.3, as it is easily seen that $y_{(-)}$ is a strict family if and only if the morphism

$$x \mapsto (x, y_x)\colon X \longrightarrow \sum_{x\,:\,X} Y_x$$

is strict.

Next we use Definition 4.4 to derive a natural notion of multistrict dependent family, and we generalise Lemma 4.2 to apply to such families.

Definition 4.5 (Multistrict family) Given an internal family $\{(Y_{x_1\ldots x_k}, \beta_{x_1\ldots x_k})\}_{x_1\,:\,X_1,\ldots,x_k\,:\,X_k}$ of pointed objects, where $(X_1,\alpha_1),\ldots,(X_k,\alpha_k)$ are pointed, we say that

$$y_{(-)\ldots(-)}\colon \prod_{x_1\,:\,X_1} \cdots \prod_{x_k\,:\,X_k} Y_{x_1\ldots x_k}$$

is a *k-strict family* if, for each i with $1 \le i \le k$, it holds in $\mathbf{C}$ that

$$\forall x_1\colon X_1,\ldots,x_{i-1}\colon X_{i-1},\ x_{i+1}\colon X_{i+1},\ldots,x_k\colon X_k.$$
$$x_i \mapsto y_{x_1\ldots x_k} \text{ is a strict family in } \prod_{x_i\,:\,X_i} Y_{x_1\ldots x_k}.$$

Lemma 4.6 *For internal families*

$$\{(Y_{i_1\ldots i_k}, \beta_{i_1\ldots i_k})\}_{i_1\,:\,\mathbf{I},\ldots,i_k\,:\,\mathbf{I}},$$
$$\{f_{i_1\ldots i_k}\colon Y_{i_1\ldots i_k} \to Y_{si_1\ldots si_k}\}_{i_1\,:\,\mathbf{I},\ldots,i_k\,:\,\mathbf{I}},$$

of pointed objects and functions respectively, there exists a unique k-strict family

$$y_{(-)\ldots(-)}\colon \prod_{i_1\,:\,\mathbf{I}} \cdots \prod_{i_k\,:\,\mathbf{I}} Y_{i_1\ldots i_k}$$

satisfying $y_{si_1\ldots si_k} = f_{i_1\ldots i_k}(y_{i_1\ldots i_k})$.

This lemma will be crucial in Section 6.

5. The limit-colimit coincidence

One of the main tools in the proof of Theorem 1 will be a variant of the limit-colimit coincidence of domain theory. The standard domain-theoretic version of this coincidence uses $\mathbf{N}$-indexed diagrams of *embedding-projection* pairs, see e.g. [32]. We wish to establish an analogous coincidence for internal categories in $\mathbf{C}$. For this, we have to make two important modifications. First, as motivated in Section 3, the diagram must be indexed by $\mathbf{I}$ rather than by $\mathbf{N}$. Second, we have to manage without any notion of embedding-projection pair. Instead, the use of $\mathbf{I}$ as an indexing object miraculously enables us to prove the limit-colimit coincidence for arbitrary diagrams satisfying some simple equational properties.

Let K be an internal category in $\mathbf{C}$. For this entire section, we reason internally in $\mathbf{C}$ about K. As we do not require K to be locally small, we refer to $\{\mathsf{K}(A,B)\}_{A,B\in|\mathsf{K}|}$ as the family of *hom-classes*.

An $\mathbf{I}$-*bichain* in K is given by families,

$$A_{(-)}\colon |\mathsf{K}|^{\mathbf{I}}$$
$$x_{(-)(-)}\colon \prod_{i\,:\,\mathbf{I}} \prod_{j\,:\,\mathbf{I}} \mathsf{K}(A_i, A_j),$$

satisfying the equations

$$x_{ii} = \mathsf{id}_{A_i} \tag{6}$$
$$x_{jk} \circ x_{ij} = x_{min(i,j,k)\ k} \circ x_{i\ min(i,j,k)}. \tag{7}$$

Here $min(i,j,k)$ means $min(i, min(j,k))$, using the operation from Section 4. Equations (6) and (7) have useful consequences relating $x_{(-)(-)}$ to the partial order $\sqsubseteq$ on $\mathbf{I}$.

Lemma 5.1 *For any $i,j,k\colon \mathbf{I}$, if $i \sqsubseteq j$ then $x_{jk} \circ x_{ij} = x_{ik}$ and $x_{ji} \circ x_{kj} = x_{ki}$, so, in particular, $x_{ji} \circ x_{ij} = \mathsf{id}_{A_i}$.*

Thus if $i \sqsubseteq j$ then x_{ij} and x_{ji} form a section-retraction pair. The limit-colimit coincidence will relate the colimit of the diagram of sections to the limit of the diagram of retractions.

Given an $\mathbf{I}$-bichain, $(A_{(-)}, x_{(-)(-)})$, we write $(x_{ij})_{i\sqsubseteq j}$ for the evident partially-ordered diagram of shape $(\mathbf{I}, \sqsubseteq)$, consisting entirely of sections. The notion of *cocone* and *colimit* for such diagrams are defined as usual. Dually, we write $(x_{ij})_{i \sqsupseteq j}$ for the evident partially-ordered diagram of shape $(\mathbf{I}, \sqsupseteq)$, consisting of retractions. The notion of *cone* and *limit* are defined as usual.

Lemma 5.2 *If X is complete then there exists a unique map $\bigsqcup : X^{\mathbf{I}} \longrightarrow X$ satisfying*

$$\mathbf{C} \models \forall x_{(-)} : X^{\mathbf{I}}.\ \forall i : \mathbf{I}.\ x_i = \bigsqcup_j x_{min(i,j)},$$

where $\bigsqcup_j x_{min(i,j)}$ means $\bigsqcup(j \mapsto x_{min(i,j)})$.

Proposition 2 (Limit-colimit coincidence)
If K is an internal category in which all hom-classes are complete then, for any $\mathbf{I}$-bichain $(A_{(-)}, x_{(-)(-)})$ in K, the following statements are equivalent.

1. *B is a limiting object for $(x_{ij})_{i \sqsupseteq j}$.*
2. *There exist a cone $l_{(-)} : \prod_{i : \mathbf{I}} \mathsf{K}(B, A_i)$ for $(x_{ij})_{i \sqsupseteq j}$ and cocone $c_{(-)} : \prod_{i : \mathbf{I}} \mathsf{K}(A_i, B)$ for $(x_{ij})_{i \sqsubseteq j}$ such that:*

$$\text{for all } i, j : \mathbf{I}, \text{ it holds that } l_j \circ c_i = x_{ij}, \qquad (8)$$

$$\bigsqcup_i (c_i \circ l_i) = \mathsf{id}_B. \qquad (9)$$

3. *B is a colimiting object for $(x_{ij})_{i \sqsubseteq j}$.*

Moreover, if 2 holds then $l_{(-)}$ is a limiting cone and $c_{(-)}$ is a colimiting cone. Furthermore, (8) and (9) together imply that each of $l_{(-)}$ and $c_{(-)}$ determines the other.

In view of the proposition, we shall henceforth refer to $(B, l_{(-)}, c_{(-)})$ satisfying (8) and (9) as a *bilimit* of the $\mathbf{I}$-bichain $(A_{(-)}, x_{(-)(-)})$.

6. Conditions for algebraic compactness

In this section we define a notion of *suitable* internal category—one satisfying conditions that are sufficient for algebraic compactness to hold. These conditions are convenient for establishing the algebraic compactness of specific internal categories, e.g. pP.

Definition 6.1 (Suitable category) A *suitable* category is given by an internal category K together with a pointed structure $(|\mathsf{K}|, \alpha)$ and a family of pointed structures $\{(\mathsf{K}(A, B), \beta_{A,B})\}_{A,B : |\mathsf{K}|}$ satisfying: for all $A, B : |\mathsf{K}|$, the hom-class $\mathsf{K}(A, B)$ is complete; for all $A, B, C : |\mathsf{K}|$, the composition function $\mathsf{K}(B, C) \times \mathsf{K}(A, B) \to \mathsf{K}(A, C)$ is bistrict; the family $\mathsf{id}_{(-)} : \prod_{A : |\mathsf{K}|} \mathsf{K}(A, A)$ is strict; and every $\mathbf{I}$-bichain in K has a specified bilimit.

In this definition, by having a specified bilimit we mean that bilimits are given by a morphism $Bichains_{\mathsf{K}} \longrightarrow Bicones_{\mathsf{K}}$ in $\mathbf{C}$, where $Bichains_{\mathsf{K}}$ is the class of standard $\mathbf{I}$-bichains in K and $Bicones_{\mathsf{K}}$ is the class of cone/cocone tuples $(B, l_{(-)}, c_{(-)})$ for $\mathbf{I}$-bichains.

The next result is the reason for introducing the notion of suitable category.

Proposition 3 *Every suitable internal category is algebraically compact.*

To prove Proposition 3, let K be a suitable category. The notion of suitable category is stable under slicing of $\mathbf{C}$, thus it suffices to show that Definition 3.1 applies in the special case of a singleton family. Accordingly, let F be an internal endofunctor on K.

As $(|\mathsf{K}|, \alpha)$ is pointed, there is, by Lemma 4.2, a unique strict map $F^{(-)}0 : \mathbf{I} \to |\mathsf{K}|$ such that $F(F^i 0) = F^{si} 0$. Here the notation is to convey the idea that one should think of $F^i 0$ as the i-th iterate of F applied to a zero object 0 in K. However, this intuition is subject to two caveats: firstly i comes from $\mathbf{I}$ rather than from $\mathbf{N}$, so the notion of iterate is non-standard; secondly, we do not yet know that K has a zero object, although the existence of one will, in the end, follow from Proposition 3, once proven.

As each $(\mathsf{K}(A, B), \beta_{A,B})$ is pointed, there exists, by Lemma 4.6, a unique bistrict family

$$x_{(-)(-)} : \prod_{i : \mathbf{I}} \prod_{j : \mathbf{I}} \mathsf{K}(F^i 0, F^j 0)$$

satisfying $x_{si\,sj} = F(x_{ij})$.

Lemma 6.2 $(F^{(-)}0, x_{(-)(-)})$ *is an $\mathbf{I}$-bichain.*

The proof is a straightforward application of Lemma 4.1 and Lemma 4.6 in the cases $k = 1, 3$.

Now we are in a position to construct the bifree algebra for F. Accordingly, let $(B, l_{(-)}, c_{(-)})$ be the specified bilimit of $(F^{(-)}0, x_{(-)(-)})$. Define a morphism $FB \xrightarrow{b} B$ by $b = \bigsqcup_i (c_{si} \circ F l_i)$.

Lemma 6.3 (B, b) *is a bifree F-algebra.*

The proof is by establishing that $FB \xrightarrow{b} B$ is a *special-F-invariant object* in the sense of [6, 29], and that this property is characteristic of bifree F-algebras, again see [6, 29]. This concludes the proof of Proposition 3.

We now complete the proof of Theorem 1 by establishing the result below.

Proposition 4 *If Axiom* 1 *holds then the internal category* pP *is suitable.*

The proof of this proposition is very long. In this conference version of the paper, we just state the non-obvious lemmas, all of which **assume Axiom 1**.

Lemma 6.4 *The morphism* $\bigcup\colon \mathbb{L}(\mathcal{P}_{\mathbf{S}}U) \longrightarrow \mathcal{P}_{\mathbf{S}}U$ *restricts to a morphism* $\bigcup\colon \mathbb{L}|\mathsf{pP}| \longrightarrow |\mathsf{pP}|$*, giving a pointed structure* $(|\mathsf{pP}|, \bigcup)$.

The proof uses [30, Lemma 6], which gives a useful interpolation condition for establishing well-completeness.

Lemma 6.5 *The internal functor* $L : \mathsf{pP} \to \mathsf{P}$ *creates (up to isomorphism) limits for diagrams of shape* $(\mathbf{I}, \sqsupseteq)$.

Corollary 6.6 pP *has bilimits of* $\mathbf{I}$*-bichains.*

7. An internal interpretation of FPC

In this section, we apply Theorem 1 to obtain an interpretation of Plotkin's call-by-value recursively typed λ-calculus, FPC, in the internal category pP.

We give a brief summary of the language FPC, introduced in [24]. For full details see [2]. We use $X, Y, \ldots$ to range over type variables, and $\sigma, \tau, \ldots$ to range over types, which are given by:

$$\sigma \quad ::= \quad X \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau \mid \mu X.\sigma.$$

Here the prefix μX binds X. We use $\Theta, \ldots$ to range over finite sequences of distinct type variables. We write $\Theta \vdash \sigma$ to mean that all free type variables in σ appear in Θ.

We use $x, y, \ldots$ to range over term variables, and $s, t, \ldots$ to range over terms, which are given by:

$$\begin{aligned} t ::= x \mid \mathtt{inl}(t) \mid \mathtt{inr}(t) \mid \mathtt{case}(s)\ \mathtt{of}\ x.t\ \mathtt{or}\ y.u \mid (s,t) \mid \\ \mathtt{fst}(t) \mid \mathtt{snd}(t) \mid \lambda x.\, t \mid s(t) \mid \mathtt{intro}(t) \mid \mathtt{elim}(t), \end{aligned}$$

where, to ease clutter, we omit certain necessary type information from $\mathtt{inl}(t)$, $\mathtt{inr}(t)$, $\mathtt{intro}(t)$ and $\lambda x.\, t$, see [2]. We use $\Gamma, \ldots$ to range over sequences of the form $x_1\colon \sigma_1, \ldots, x_k\colon \sigma_k$ with all x_i distinct and all σ_i closed. For closed types σ, we write $\Gamma \vdash t\colon \sigma$ to mean that t is a well-formed term of type σ relative to Γ, where the rules for deriving such typing assertions are as in [2].

To define a call-by-value operational semantics for FPC, we first specify the *values*, closed terms $v, \ldots$ of the form:

$$v \quad ::= \quad \mathtt{inl}(v) \mid \mathtt{inr}(v) \mid (v_1, v_2) \mid \lambda x.\, t \mid \mathtt{intro}(v).$$

The call-by-value evaluation relation $t \rightsquigarrow v$ between closed terms t and values v is defined as in [2]. We say that a closed term t *converges*, notation $t \Downarrow$, if there exists (a necessarily unique) v such that $t \rightsquigarrow v$.

To interpret FPC in pP, we need pP to be closed under $+$, so henceforth in this section we **assume Axiom 2**.

First, we interpret types. To apply algebraic compactness it is necesary to interpret open types as internal functors. Moreover, because of the bivariance of $\rightharpoonup$, they must be interpreted as internal functors on the internal category $\mathsf{pP}^{op} \times \mathsf{pP}$, for which we write $\widehat{\mathsf{pP}}$. The functors will all be symmetric in the sense of [2, §6.3]. Indeed, an open type σ is interpreted, relative to any $\Theta = X_1, \ldots, X_k$ such that $\Theta \vdash \sigma$, as a symmetric internal functor,

$$(\![\Theta \vdash \sigma]\!) \quad : \quad \widehat{\mathsf{pP}}^{k} \to \widehat{\mathsf{pP}}\,.$$

The interpretation is defined by induction on the structure of σ. Type variables, sum, product and function types are easily handled using symmetric extensions of projections, $+$, $\times$ and $\rightharpoonup$ to $\widehat{\mathsf{pP}}$. The definition for recursive types is

$$(\![\Theta \vdash \mu X.\sigma']\!) \;=\; (\![\Theta, X \vdash \sigma']\!)^{\dagger}\,,$$

using Lemma 3.2. (It follows from the construction of bifree algebras in suitable categories that $(\![\Theta, X \vdash \sigma']\!)^{\dagger}$ is symmetric whenever $(\![\Theta \vdash \mu X.\sigma']\!)$ is.)

For closed types, the functor $(\![\vdash \sigma]\!)\colon 1 \to \widehat{\mathsf{pP}}$, where 1 is the terminal internal category, corresponds, by symmetry, to an object in $\widehat{\mathsf{pP}}$ of the form (A, A). We write $(\![\sigma]\!)$ for the corresponding object A of pP.

For a closed type $\mu X.\sigma$, we have that $(\![\vdash \mu X.\sigma]\!)$ carries the canonical bifree algebra structure for the symmetric functor $(\![X \vdash \sigma]\!)\colon \widehat{\mathsf{pP}} \to \widehat{\mathsf{pP}}$. The bifree algebra unpacks to give isomorphisms in pP

$$\iota_{\mu X.\sigma}\colon (\![\sigma[\mu X.\sigma/X]]\!) \longrightarrow (\![\mu X.\sigma]\!) \tag{10}$$

$$\epsilon_{\mu X.\sigma}\colon (\![\mu X.\sigma]\!) \longrightarrow (\![\sigma[\mu X.\sigma/X]]\!)\,, \tag{11}$$

where $\iota_{\mu X.\sigma} = {\epsilon_{\mu X.\sigma}}^{-1}$.

To interpret the terms of FPC in pP, a context $\Gamma = x_1\colon \sigma_1, \ldots, x_k\colon \sigma_k$ is interpreted as the object $(\![\Gamma]\!) = (\![\sigma_1]\!) \times \cdots \times (\![\sigma_k]\!)$ of pP. A term $\Gamma \vdash t\colon \sigma$ is interpreted as a morphism $(\![t]\!)_\Gamma$ from $(\![\Gamma]\!)$ to $(\![\sigma]\!)$ in pP, i.e. as a point $(\![t]\!)_\Gamma\colon \mathbf{1} \longrightarrow \mathsf{pP}((\![\Gamma]\!), (\![\sigma]\!))$ in $\mathbf{C}$. The definition of $(\![t]\!)_\Gamma$, by induction on the structure of t, is standard, see e.g. [2]. When Γ is empty (i.e. t is closed), we simply write $(\![t]\!)$.

8. Internal computational adequacy

Assuming Axiom **2**, we have interpreted FPC in the internal category pP. To proceed further, we need to formalize the the syntax of FPC and its operational semantics in $\mathbf{C}$. This is an exercise in Gödel numbering. Types and terms are thus encoded as natural numbers. We write $\mathcal{T}_\sigma$ for the object of Gödel numbers of closed terms of type σ, and $\mathcal{V}_\sigma$ for the object of Gödel numbers of closed values of type σ. Both $\mathcal{V}_\sigma$ and $\mathcal{T}_\sigma$ are primitive recursive subobjects of $\mathbf{N}$. The operational semantics is encoded so that $t \rightsquigarrow v$

and $t\Downarrow$ are both Σ^0_1 relations on Gödel numbers for t and v. For notational convenience, we choose not to make a syntactic distinction between the formalized relations and the actual relations. The meaning should always be clear from the context. For example, the statement $t\Downarrow$ in Proposition 5 below obviously uses the formalized operational semantics.

The main result of this section establishes the equivalence of operational and denotational notions of convergence, as interpreted within $\mathbf{C}$. For the denotational notion, given $\vdash\ t\colon\sigma$, we write $\llparenthesis t\rrparenthesis\downarrow$ if the point $\llparenthesis t\rrparenthesis\colon \mathbf{1} \longrightarrow \mathsf{pP}(1, \llparenthesis\sigma\rrparenthesis)$ gives a total function in $\mathsf{pP}(1, \llparenthesis\sigma\rrparenthesis) = 1 \rightharpoonup \llparenthesis\sigma\rrparenthesis$.

Proposition 5 (Internal computational adequacy) *If Axiom $\mathbf{N}$ holds then, for all terms $\vdash\ t\colon\sigma$, it holds that $\mathbf{C}\ \models\ t\Downarrow \leftrightarrow\ \llparenthesis t\rrparenthesis\downarrow$.*

The implication $t\Downarrow\ \rightarrow\ \ \llparenthesis t\rrparenthesis\downarrow$ is easily proved by induction on the derivation of the evaluation relation for t, and does not require Axiom $\mathbf{N}$. For the proof of the converse implication, we adapt the approach of [24, 2] to our setting.

The strategy is to define binary relations relating closed terms to their internal denotations. A closed term $t\colon\sigma$ has a denotation $\llparenthesis t\rrparenthesis\colon \mathsf{pP}(1, \llparenthesis\sigma\rrparenthesis)$. However, values $v\colon\sigma$ enjoy the extra property that $\llparenthesis v\rrparenthesis\downarrow$, i.e. that $\llparenthesis v\rrparenthesis\colon \mathsf{P}(1, \llparenthesis\sigma\rrparenthesis)$, using the hom-set inclusion given by

$$\mathsf{P}(A,B)\ =\ \{f\colon \mathsf{pP}(A,B) \mid f \text{ is total}\} \rightarrowtail \mathsf{pP}(A,B)\,,$$

which holds for any $A, B\colon|\mathsf{P}|$ (equivalently $A,B\colon|\mathsf{pP}|$). For each closed type σ, we define binary relations in $\mathbf{C}$,

$$\begin{array}{rcl} \preceq_\sigma & \rightarrowtail & \mathsf{P}(1, \llparenthesis\sigma\rrparenthesis)\times\mathcal{V}_\sigma \\ \precsim_\sigma & \rightarrowtail & \mathsf{pP}(1, \llparenthesis\sigma\rrparenthesis)\times\mathcal{T}_\sigma\,, \end{array}$$

related to each other by

$$e \precsim_\sigma t\ \text{ iff }\ e\downarrow \text{ implies } \exists v\colon\mathcal{V}_\sigma.\ t \rightsquigarrow v \text{ and } e \preceq_\sigma v\,, \quad (12)$$

making use of the operational semantics as formalized in $\mathbf{C}$.

The relations $\preceq_\sigma$ and $\precsim_\sigma$ are defined so that they satisfy the following internal equivalences in $\mathbf{C}$.

$$\begin{array}{rcl}
d \preceq_{\sigma+\tau} \mathtt{inl}(v) & \text{iff} & d = \mathrm{inl}(c) \text{ where } c \preceq_\sigma v \\
d \preceq_{\sigma+\tau} \mathtt{inr}(v) & \text{iff} & d = \mathrm{inr}(c) \text{ where } c \preceq_\tau v \\
(c,d) \preceq_{\sigma\times\tau} (u,v) & \text{iff} & c \preceq_\sigma u \text{ and } d \preceq_\tau v \\
f \preceq_{\sigma\to\tau} \lambda x.\, t & \text{iff} & \forall\, d\colon \mathsf{P}(1, \llparenthesis\sigma\rrparenthesis),\ \forall\, v\colon \mathcal{T}_\sigma. \\
 & & \qquad d \preceq_\sigma v\ \rightarrow\ f(d) \precsim_\tau t[v/x] \\
d \preceq_{\mu X.\sigma} \mathtt{intro}(v) & \text{iff} & \epsilon_{\mu X.\sigma}(d) \preceq_{\sigma[\mu X.\sigma/X]} v
\end{array}$$

The clauses for the non-recursive types apply the internal partial bicartesian-closed structure $(+, \times, \rightharpoonup)$ of pP, using a self-explanatory notation. The clause for $\mu X.\sigma$ involves the isomorphism (11), which, because it is an isomorphism in pP, is also an isomorphism in P.

Because the above relations are recursively specified, constructing them takes considerable work. They can be obtained by adapting the approach of [24, 2]. Very briefly, for each closed σ, we define an internal category R_σ. Internally in $\mathbf{C}$, objects are pairs $R = (|R|, \preceq_R)$ satisfying,

1. $|R|\colon \mathcal{P}_{\mathbf{S}}U$ is a predomain,
2. $\preceq_R$ is a binary relation between $\mathsf{P}(1, |R|)$ and $\mathcal{V}_\sigma$,
3. for all $v\colon\mathcal{V}_\sigma$, $\{d\colon \mathsf{P}(1,|R|) \mid d \preceq_R v\}$ is a preomain.

Morphisms are partial functions preserving the relations. One proves that each category R_σ is suitable, hence algebraically compact. Following [24, 2], the relations $\preceq_\sigma$ are then constructed by defining a non-standard interpretation of types in the R_σ categories, using relational "liftings" of the functors $+$, $\times$ and $\rightharpoonup$, on pP, to the R_σ categories. Axiom $\mathbf{N}$, in the guise of Lemma 2.4.7, is used crucially in obtaining the relational lifting of $\rightharpoonup$.

Once the relations have been defined, the lemma below can be established by a straightforward (external) induction on the structure of t.

Lemma 8.1 *If $x_1\colon\tau_1, \ldots, x_k\colon\tau_k \vdash t\colon\sigma$ then*

$$\begin{array}{l}
\mathbf{C}\ \models \forall\, d_1\colon \mathsf{P}(1, \llparenthesis\tau_1\rrparenthesis),\ \ldots\ , d_k\colon \mathsf{P}(1, \llparenthesis\tau_k\rrparenthesis), \\
\qquad \forall\, v_1\colon \mathcal{V}_{\tau_1},\ \ldots\ , v_k\colon\mathcal{V}_{\tau_k}. \\
\qquad\quad d_1 \preceq_{\tau_1} v_1\ \wedge\ \ldots\ \wedge\ d_k \preceq_{\tau_k} v_k\ \rightarrow \\
\qquad\qquad \llparenthesis t\rrparenthesis_\Gamma(d_1, \ldots, d_k) \precsim_\sigma t[v_1, \ldots, v_k/x_1, \ldots, x_k].
\end{array}$$

To show that Proposition 5 follows from Lemma 8.1, take any closed term $t\colon\sigma$. Then, by the lemma, $\mathbf{C} \models \llparenthesis t\rrparenthesis \precsim_\sigma t$. Hence, by (12), $\mathbf{C} \models\ \llparenthesis t\rrparenthesis\downarrow\rightarrow (\exists v\colon\mathcal{V}_\sigma.\ t \rightsquigarrow v \wedge \llparenthesis t\rrparenthesis \preceq_\sigma v)$. So indeed $\mathbf{C} \models \llparenthesis t\rrparenthesis\downarrow\rightarrow\ t\Downarrow$.

9 An external interpretation of FPC

So far, we have given an *internal* interpretation of FPC, in the internal category pP. We now extract an *external* "real world" interpretation in the category $\mathbf{pP}$. A closed type σ is interpreted as an object $\llbracket\sigma\rrbracket$ of $\mathbf{pP}$, by defining $\llbracket\sigma\rrbracket$ as the pullback below.

$$\begin{tikzcd}
\llbracket\sigma\rrbracket \arrow[rr] \arrow[d] \arrow[drr, phantom, "\lrcorner", very near start] & & \ni_U \arrow[d, "\gamma_U"] \\
\mathbf{1} \arrow[r, "\llparenthesis\sigma\rrparenthesis"'] & {|\mathsf{pP}|} \arrow[r, tail] & \mathcal{P}_{\mathbf{S}}U\,,
\end{tikzcd}$$

where γ_U is as in (1) from Section 2. The object $\llbracket\sigma\rrbracket$ is indeed a predomain by the definition of $|\mathsf{pP}|$ as a subobject of $\mathcal{P}_{\mathbf{S}}U$. Similarly, a context Γ is interpreted as an object $\llbracket\Gamma\rrbracket$, by replacing σ with Γ in the diagram above. We interpret a term $\Gamma \vdash t\colon \sigma$ as a morphism $[\![t]\!]_\Gamma\colon [\![\Gamma]\!] \longrightarrow [\![\sigma]\!]$ in $\mathbf{pP}$, by transposing $(\![t]\!)_\Gamma\colon \mathsf{pP}((\![\Gamma]\!), (\![\sigma]\!))$ in the evident way. When Γ is empty, we write simply $[\![t]\!]\colon \mathbf{1} \longrightarrow [\![\sigma]\!]$. We write $[\![t]\!]\downarrow$ if the partial map $[\![t]\!]$ is total.

Our last main result is a complete characterisation, in terms of a property of the internal logic of $\mathbf{C}$, of when the external interpretation of FPC is computationally adequate. Using the natural numbers object $\mathbf{N}$, one can define a standard encoding of any primitive recursive predicate as a (decidable) subobject $P \rightarrowtail \mathbf{N}$. A Σ^0_1-sentence is a statement of the form $\exists n\colon \mathbf{N}.\, P(n)$ where P is a primitive recursive predicate. We say that $\mathbf{C}$ is *1-consistent* if, for every Σ^0_1-sentence φ, $\mathbf{C} \models \varphi$ implies that ϕ is true in reality.

Theorem 2 (External computational adequacy) *If Axiom $\mathbf{N}$ holds then the following are equivalent.*

1. *For all $\vdash t\colon \sigma$, it holds that $t\Downarrow$ if and only if $[\![t]\!]\downarrow$, i.e. the interpretation is computationally adequate.*

2. *$\mathbf{C}$ (equivalently $\mathbf{S}$) is 1-consistent.*

An in [30, Corollary 1], a consequence of the theorem is that there exist categories with class structure, $\mathbf{C}$, satisfying Axiom $\mathbf{N}$, for which the interpretation of FPC in $\mathbf{pP}$ is not computationally adequate. However, such categories $\mathbf{C}$ are pathologies. Instead, the main force of Theorem 2 is in the converse implication, which reduces computational adequacy to a very weak and ubiquitous condition. This will be exploited in Section 10.

We only briefly outline the proof of Theorem 2. To prove that computational adequacy implies 1-consistency, we use the standard encoding of the type of natural numbers as the FPC type $\mu Y.\,(\mu X.X \to \mu X.X) + Y$. The usual primitives on natural numbers are easily defined. Also FPC supports the recursive definition of functions, see e.g. [2]. Thus one can interpret call-by-value PCF in FPC. The proof that computational adequacy implies 1-consistency, can now be borrowed from [30, §6]. The simple idea is to encode any Σ^0_1-sentence as a search program that terminates if and only if the sentence is true.

The converse implication, that 1-consistency implies computational adequacy, follows swiftly from Proposition 5. For the interesting implication, suppose that $[\![t]\!]\downarrow$, or equivalently $\mathbf{C} \models (\![t]\!)\downarrow$. By Proposition 5, $\mathbf{C} \models t\Downarrow$. However, $t\Downarrow$ is a Σ^0_1 sentence, so if $\mathbf{C}$ is 1-consistent then indeed $t\Downarrow$.

10. Applications

10.1. Realizability models

A realizability model is specified by a partial combinatory algebra $(A, \cdot)$, which determines a category $\mathbf{Mod}(A)$ of *modest sets* over A, see e.g. [17, §2–3]. In many such categories, one can find a dominance Σ, often conveniently determined by a *divergence* $D \subset A$ (see [17, Def. 4.1]), such that Axiom $\mathbf{2}$ holds. Numerous examples are presented in [16, 17]. Furthermore, by [17, Theorem 7.5], it follows that Axiom $\mathbf{N}$ holds.

As is well-known, there is a full embedding

$$\mathbf{Mod}(A) \hookrightarrow \mathbf{RT}(A)$$

of modest sets into the *realizability topos* over A [10, 12]. Assuming a strongly inaccessible cardinal, one can follow [15, §IV.4] and endow $\mathbf{RT}(A)$ with class structure. By constructing the initial ZF-algebra V in $\mathbf{RT}(A)$, and then applying [31, Theorem 7], one extracts a full subcategory $\mathbf{RT}_{<V}(A) \hookrightarrow \mathbf{RT}(A)$, with class structure, in which V is a universal object. This category contains the dominance Σ, and inherits Axiom $\mathbf{N}$ from $\mathbf{Mod}(A)$. Thus the results of this paper can be applied to obtain a category of predomains $\mathbf{P} \hookrightarrow \mathbf{RT}_{<V}(A)$ in which FPC can be interpreted. Moreover, it can be shown that the interpretation of FPC lives within the subcategory $\mathbf{Mod}(A) \hookrightarrow \mathbf{RT}_{<V}(A)$.

If A is nontrivial then the category $\mathbf{RT}_{<V}(A)$ is 1-consistent, because its numerals are standard. Thus, by Theorem 2, the interpretation of FPC in $\mathbf{Mod}(A)$ is computationally adequate. This gives the first proof of computational adequacy for the interpretation of FPC in the realizability models of [9, 20, 21, 22, 16, 17].

10.2. Models of axiomatic domain theory

In [2], an axiomatization of a general order-enriched notion of model for FPC is given, and computational adequacy is proved for any nontrivial model satisfying an additional *absoluteness* condition. In [4, 3], a much more general class of enriched models is introduced, although the interpretation of FPC is not explicitly considered. Following the approach of [3], we can accommodate many of the models of [2, 4, 3] within our setting.

Let $\mathcal{C}$ be any *lifting monadic enrichment base* in the sense of [3, Def. 1.12]. In particular, $\mathcal{C}$ has dominance Σ and a lifting functor $\mathbb{L}$. We assume further that $\mathcal{C}$ has stable countable coproducts. The Yoneda functor gives a full embedding

$$\mathbf{y}\colon \mathcal{C} \hookrightarrow \mathit{Sh}(\mathcal{C}, \mathsf{Can})\,,$$

where $\mathit{Sh}(\mathcal{C}, \mathsf{Can})$ is the category of sheaves for the canonical Grothendieck topology Can on $\mathcal{C}$. It holds that $\mathbf{y}(\Sigma)$ is a dominance in $\mathit{Sh}(\mathcal{C}, \mathsf{Can})$, and, because $\mathcal{C}$ has stable countable coproducts, Axiom $\mathbf{N}$ is satisfied (n.b. coproducts are automatically disjoint by [2, Prop. 5.3.12]).

Assuming a strongly inaccessible cardinal, one can follow [15, §IV.3] and endow $\mathit{Sh}(\mathcal{C}, \mathsf{Can})$ with class structure. As before, by constructing the initial ZF-algebra V in $Sh(\mathcal{C}, \mathsf{Can})$, one extracts a full subcategory $Sh_{<V}(\mathcal{C}, \mathsf{Can}) \hookrightarrow Sh(\mathcal{C}, \mathsf{Can})$, with class structure, in which V is a universal object. This category contains the dominance $\mathbf{y}(\Sigma)$, and inherits Axiom $\mathbf{N}$ from $Sh(\mathcal{C}, \mathsf{Can})$. Thus the results of this paper can be applied to obtain a category of predomains $\mathbf{P} \hookrightarrow Sh_{<V}(\mathcal{C}, \mathsf{Can})$ in which FPC can be interpreted. Furthemore, under the condition that $\mathcal{C}$ is a *KADT model* [3, Def. 1.12], it can be shown that the interpretation of FPC lives within the subcategory $\mathcal{C} \hookrightarrow Sh_{<V}(\mathcal{C}, \mathsf{Can})$.

In any nontrivial Grothendieck topos, internal first-order arithmetic is simply classical true arithmetic. Thus $Sh(\mathcal{C}, \mathsf{Can})$ is 1-consistent and hence so is $Sh_{<V}(\mathcal{C}, \mathsf{Can})$. Therefore the interpretation of FPC in any KADT model with stable countable coproducts is computationally adequate. As a special case, we obtain that the interpretation of FPC in any nontrivial *domain-theoretic model* (in the sense of [2, §8.5.1]) with stable countable coproducts is computationally adequate. Thus we replace the *absoluteness* condition for computational adequacy in [2] with the apparently incomparable requirement of stable countable coproducts. More strikingly, we also have the first computational adequacy result that applies to the more general class of enriched models considered in [4, 3].

References

[1] R. Amadio. On the adequacy of PER models. In *Mathematical Foundations of Computer Science 1993*, pages 222–431. Springer LNCS 711, 1993.

[2] M.P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps*. Distinguished Dissertation Series, CUP, 1996.

[3] M.P. Fiore and G.D. Plotkin. An extension of models of axiomatic domain theory to models of synthetic domain theory. In *Proceedings of CSL 96*, pages 129–149. Springer LNCS 1258, 1997.

[4] M.P. Fiore, G.D. Plotkin, and A.J. Power. Complete cuboidal sets in axiomatic domain theory. In *Proc. 12th Annual Symposium on Logic in Computer Science*, pages 268–279, 1997.

[5] M.P. Fiore and G. Rosolini. Domains in H. *Theoretical Computer Science*, 264:171–193, 2001.

[6] P.J. Freyd. Recursive types reduced to inductive types. In *Proc. 5th Annual Symposium on Logic in Computer Science*, pages 498 – 507, 1990.

[7] P.J. Freyd. Algebraically complete categories. In *Category Theory, Proc. Como 1990*, Springer LNM 1488, pages 95–104, 1991.

[8] P.J. Freyd. Remarks on algebraically compact categories. In *Applications of Categories in Computer Science*, pages 95–106. LMS Lecture Notes 177, CUP, 1992.

[9] P.J. Freyd, P. Mulry, G. Rosolini, and D.S. Scott. Extensional PERs. *Information and Computation*, 98:211–227, 1992.

[10] J.M.E. Hyland. The effective topos. In *L.E.J. Brouwer Centenary Symposium*, pages 165–216. North-Holland, 1982.

[11] J.M.E. Hyland. First steps in synthetic domain theory. In *Category Theory, Proc. Como 1990*, pages 131–156. Springer LNM 1488, 1991.

[12] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980.

[13] B. Jacobs. *Categorical Logic and Type Theory*. North Holland, Amsterdam, 1999.

[14] M. Jibladze. A presentation of the initial lift algebra. *Journal of Pure and Applied Algebra*, 116:185–198, 1997.

[15] A. Joyal and I. Moerdijk. *Algebraic Set Theory*. LMS Lecture Notes 220, CUP, 1995.

[16] J.R. Longley. *Realizability Toposes and Language Semantics*. Ph.D. thesis, Department of Computer Science, University of Edinburgh. Available as ECS-LFCS-95-332, 1995.

[17] J.R. Longley and A.K. Simpson. A uniform account of domain theory in realizability models. *Math. Struct. in Comp. Sci.*, 7:469–505, 1997.

[18] G. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertation Series, Springer-Verlag, 1998.

[19] J. van Oosten and A.K. Simpson. Axioms and (counter)-examples in synthetic domain theory. *Annals of Pure and Applied Logic*, 104:233–278, 2000.

[20] W.K.-S. Phoa. Effective domains and intrinsic structure. In *Proc. 5th Annual Symposium on Logic in Computer Science*, pages 366–377, 1990.

[21] W.K.-S. Phoa. Building domains from graph models. *Math. Struct. in Comp. Sci.*, 2:277–299, 1992.

[22] W.K.-S. Phoa. From term models to domains. *Information and Computation*, 109:211–255, 1994.

[23] G.D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[24] G.D. Plotkin. *Denotational semantics with partial functions*. Lecture notes, C.S.L.I. Summer School, 1985.

[25] B. Reus and Th. Streicher. General synthetic domain theory — a logical approach. *Math. Struct. in Comp. Sci.*, 9:177–223, 1999.

[26] G. Rosolini. *Continuity and Effectivity in Topoi*. PhD thesis, University of Oxford, 1986.

[27] A. Ščedrov. Intuitionistic set theory. In *Harvey Friedman's Research on The Foundations of Mathematics*, pages 257–184. Elsevier Science Publishers, 1985.

[28] D.S. Scott. Relating theories of the λ-calculus. In *To H.B. Curry*, pages 403–450. Academic Press, 1980.

[29] A.K. Simpson. *Recursive types in Kleisli categories*. Unpublished manuscript, University of Edinburgh, 1992.

[30] A.K. Simpson. Computational adequacy in an elementary topos. In *Computer Science Logic, Proceedings CSL '98*, pages 323–342. Springer LNCS 1584, 1999.

[31] A.K. Simpson. Elementary axioms for categories of classes. In *Proc. 14th Annual Symposium on Logic in Computer Science*, pages 77–85, 1999.

[32] M.B. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal of Computing*, 11:761–783, 1982.

[33] P. Taylor. The fixed point property in synthetic domain theory. In *Proc. 6th Annual Symposium on Logic in Computer Science*, pages 152–160, 1991.

[34] P. Taylor. *Practical Foundations*. Cambridge University Press, 1999.