Domain-theoretic models of parametric polymorphism

L. Birkedal[*]
IT University of Copenhagen

R.E. Møgelberg[†]
DISI, Università di Genova

R.L. Petersen
IT University of Copenhagen

April 27, 2007

Abstract

We present a domain-theoretic model of parametric polymorphism based on admissible per's over a domain-theoretic model of the untyped lambda calculus. The model is shown to be a model of Abadi & Plotkin's logic for parametricity, by the construction of an LAPL-structure as defined by the authors in [7, 5]. This construction gives formal proof of solutions to a large class of recursive domain equations, which we explicate. As an example of a computation in the model, we explicitly describe the natural numbers object obtained using parametricity.

The theory of admissible per's can be considered a domain theory for (impredicative) polymorphism. By studying various categories of admissible and chain complete per's and their relations, we discover a picture very similar to that of domain theory.

1 Introduction

In this paper we show how to define parametric domain-theoretic models of polymorphic intuitionistic / linear lambda calculus. The work is motivated by two different observations, due to Reynolds and Plotkin.

In 1983 Reynolds argued that parametric models of the second-order lambda calculus are very useful for modeling data abstraction in programming [23] (see also [17] for a recent textbook description). For real programming, one is of course not just interested in a strongly terminating calculus such as the second-order lambda calculus, but also in a language with full recursion. Thus in *loc. cit.* Reynolds also asked for a parametric *domain-theoretic* model of polymorphism. Informally, what is meant [24] by this is a model of an extension of the polymorphic lambda calculus [22, 9], with a polymorphic fixed-point operator $Y : \forall\alpha.\, (\alpha \to \alpha) \to \alpha$ such that

1. types are modeled as domains, the sublanguage without polymorphism is modeled in the standard way and $Y\sigma$ is the least fixed-point operator for the domain σ;
2. the logical relations theorem (also known as the abstraction theorem) is satisfied when the logical relations are admissible, i.e., strict and closed under limits of chains;

[*]Corresponding author. Address: L. Birkedal, IT University of Copenhagen, Rued Langgaardsvej 7, DK–2300 Copenhagen S, DENMARK, `birkedal@itu.dk`

[†]This work is sponsored by Danish Research Agency stipend no. 272-05-0031

3. every value in the domain representing some polymorphic type is parametric in the sense that it satisfies the logical relations theorem (even if it is not the interpretation of any expression of that type).

Of course, this informal description leaves room for different formalizations of the problem. Even so, it has proved to be a non-trivial problem. Unpublished work of Plotkin [20] indicates one way to solve the problem model-theoretically by using strict, admissible partial equivalence relations over a domain model of the untyped lambda calculus but the details of this relationally parametric model have not been worked out before. We do that here.

In *loc. cit.* Plotkin also suggested that one should consider parametric domain-theoretic models not only of polymorphic lambda calculus but of polymorphic intuitionistic / linear lambda calculus. This is necessary, since full parametricity for second order lambda calculus gives a type theory with coproducts, and since we already have fixed points in the calculus, such an extension of simply typed lambda calculus is inconsistent [11]. The polymorphic intuitionistic / linear type theory gives a way to distinguish, in the calculus, between strict and possibly non-strict continuous functions and a restricted parametricity principle can then give type encodings in the linear part of the calculus. Indeed Plotkin argued that such a calculus could serve as a very powerful metalanguage for domain theory in which one could also encode recursive types, using parametricity.

Thus parametric domain-theoretic models of polymorphic intuitionistic / linear lambda calculus are of import both from a programming language perspective (for modeling data abstraction) and from a purely domain-theoretic perspective.

This paper describes such a model, classifies the class of recursive domain equations that can be solved in the model and provides the first rigorous proof that the solutions can obtained through the use of parametricity.

The proof builds on earlier work by the authors. In a recent paper [6] (see also the brief conference version [7]) we have presented an adaptation of Abadi & Plotkin's logic for parametricity for the second order lambda calculus [21] to the dual calculus suggested by Plotkin. We call this logic Linear Abadi & Plotkin Logic (LAPL), and the term language, called PILL$_Y$ for polymorphic intuitionistic / linear lambda calculus, is a simple extension of Barber and Plotkin's calculus for dual intuitionistic / linear lambda calculus (DILL) with polymorphism and fixed points. In the logic we have given detailed proofs of correctness of Plotkin's encodings of types in PILL$_Y$, including general recursive types, and also validated reasoning principles for these types.

In another recent paper [5] we have defined the category-theoretic notion of *parametric LAPL-structure*, which are parametric models of LAPL. The notion of a parametric LAPL-structure is a useful notion of parametric model since one can reason about a parametric LAPL-structure using the logic. In particular, we have shown how to solve general recursive type equations in these structures.

This paper presents a parametric PILL$_Y$-model based on admissible per's (partial equivalence relations) over a reflexive domain (a domain-theoretic model of the untyped lambda calculus) thus confirming the folklore idea that such a model exists. The model is constructed using Robinson and Rosolini's parametric completion process [25], and shown to be parametric by the construction of an LAPL-structure around it. The LAPL structure gives formal proofs of the expected consequences of parametricity. Thus by the general results for parametric LAPL-structures, we get solutions to recursive type equations; here we explicitly describe the class of recursive type equations on the model that can be solved using parametricity.

The theory of admissible per's mixes the idea of modeling impredicative polymorphism using per's with domain theory and can be seen as a domain theory for polymorphism. It is our hope that this theory will provide the same intuition about polymorphism in combination with recursion as domain theory does for the theory of recursive functions. From the view point of axiomatic domain theory, PILL$_Y$ axiomatizes the adjunction between the categories of pointed cpo's with strict continuous maps and all continuous maps respectively, whereas axiomatic domain theory traditionally has axiomatized the adjunction between the category of cpo's and the category of cpo's with partial maps (as in Fiore's thesis [8]). We see a tight correspondence to traditional domain theory and can, as usual, construct categories corresponding to pointed cpo's with strict maps and cpo's with partial maps, but unlike in traditional domain theory, the two categories are not equivalent in the setup with admissible per's.

The idea of PILL$_Y$ as a meta language for domain theory is further investigated in recent work by Møgelberg [15], in which it is shown that a large class of parametric LAPL-structures model Plotkin's FPC [19] (see also [8]) - a calculus with general recursive types and a call-by-value operational semantics. A classical result states that FPC can be interpreted in domain theory and that this model is adequate. The concrete case of the LAPL-structure investigated in this paper also models an extension of FPC with call-by-value polymorphism and this interpretation is computationally adequate.

Recently, Pitts and coworkers [18, 4] have presented a syntactic approach to Reynolds' challenge, where the notion of domain is essentially taken to be equivalence classes of terms modulo a particular notion of contextual equivalence derived from an operational semantics for a language called Lily, which is essentially polymorphic intuitionistic / linear lambda calculus endowed with an operational semantics.

In parallel with the work presented here, Rosolini and Simpson [26] have shown how to construct parametric domain-theoretic models using synthetic domain-theory in intuitionistic set-theory. Moreover, they have shown how to give a computationally adequate denotational semantics of Lily.

In subsequent papers we show how these models give rise to parametric LAPL-structures, and so the results about LAPL-structures (such as solutions to recursive domain equations) transfer to these models.

We have strived to make this paper reasonably self-contained and thus include definitions and proofs of the relevant properties for admissible per's. Moreover, we have included an overview of the concrete interpretation in Section 3.2. However, to fully appreciate the larger scope of the paper, the reader is expected to be familiar with the brief description of LAPL in the conference paper [7], but readers interested only in the description of the domain theoretic model of parametric polymorphism may skip Section 4 and consider that section a formal verification of the parametricity results for the model.

1.1 Outline

Section 2 considers two categories of admissible per's over a reflexive cpo, one with continuous maps and one with strict continuous maps. The first is shown to be cartesian closed and the second to be symmetric monoidal closed, and the two are related by an adjunction in which one map is forgetful and the other is a lifting functor. Section 2.2 contains the discussion of axiomatic domain theory advertised above.

In Section 3 a model of PILL$_Y$ in which types are indexed families of admissible per's is constructed. In Section 3.1 the parametric completion process is applied to this model giving a parametric PILL$_Y$ model. The model is parametric in the sense that it can be extended to a parametric LAPL-structure, i.e., a model for the logic LAPL for parametricity. This is shown in Section 4. Section 5 introduces the family of recursive domain equations that may be solved in the parametric model using the general results about parametric LAPL-structures, and in Section 6, as an example of a computation in the model, we compute explicitly the natural numbers object in the category of admissible per's and strict continuous maps, as encoded using parametricity. Section 7 relates our results to previous work on recursive types in per-models.

2 Admissible per's

Recall that a reflexive cpo is a pointed ω-chain-complete partial order equipped with maps

$$\Phi\colon D \to [D \to D] \qquad \text{and} \qquad \Psi\colon [D \to D] \to D,$$

both Scott-continuous and satisfying

$$\Phi \circ \Psi = id_{[D\to D]}$$

where $[D \to D]$ denotes the cpo of continuous functions from D to D. We assume, without loss of generality, that both Φ and Ψ are strict. The maps Φ, Ψ induce a combinatory algebra structure on D with application $d \cdot d' = \Phi(d)(d')$, and using this it is quite standard to construct strict continuous functions

$$\langle\cdot,\cdot\rangle\colon D \times D \to D, \qquad \pi\colon D \to D \qquad \text{and} \qquad \pi'\colon D \to D,$$

such that for all $d, d' \in D$:

$$\pi\langle d, d'\rangle = d \qquad \text{and} \qquad \pi'\langle d, d'\rangle = d'.$$

We use $\mathbf{i}$ to denote $\Psi(id_{[D\to D]})$. Notice that $\Phi(\mathbf{i}) = id_{[D\to D]}$.

Recall that a partial equivalence relation (a per) is a symmetric and transitive relation. For a per R, the set $|R|$ of elements d such that $d\ R\ d$ is called the *domain* of the per R, and R induces an equivalence relation on its domain.

Definition 2.1. An *admissible partial equivalence relation on D* is a partial equivalence relation R on D satisfying

strict $\bot_D\ R\ \bot_D$,

ω**-chain complete** For $(d_n)_{n\in\omega}$ and $(d'_n)_{n\in\omega}$ ω-chains in D:

$$(\forall n \in \omega . d_n\ R\ d'_n) \Rightarrow \bigsqcup_{n\in\omega} d_n\ R\ \bigsqcup_{n\in\omega} d'_n,$$

Definition 2.2. For R and S per's on D, define the set of **equivariant functions from R to S** as

$$\mathcal{F}(R,S) = \{\alpha \in [D \to D] | d\ R\ d' \Rightarrow \alpha(d)\ S\ \alpha(d')\}$$

and the set of **strict equivariant functions from R to S** as

$$\mathcal{F}(R,S)_\bot = \{\alpha \in \mathcal{F}(R,S) | \alpha(\bot_D) = \bot_D\}.$$

Note $\mathcal{F}(R,S)_\bot \subseteq \mathcal{F}(R,S)$.

Definition 2.3. For R and S per's on D, define on $\mathcal{F}(R,S)$ or $\mathcal{F}(R,S)_\bot$ the equivalence relation $\simeq_{R,S}$ by

$$\alpha \simeq_{R,S} \beta \Leftrightarrow \forall d \in D.\ d\ R\ d \Rightarrow \alpha(d)\ S\ \beta(d)$$

We write $\mathbf{PER}(D)$ for the category of partial equivalence relations over D. Recall that it has partial equivalence relations over D as objects and that a morphism $[\alpha]\colon R \to S$ is an equivalence class in $\mathcal{F}(R,S)/\simeq_{R,S}$. Elements of $[\alpha]$ are called **realizers** for $[\alpha]$.

Definition 2.4. We define the category $\mathbf{AP}(D)$ of admissible partial equivalence relations over D as the full subcategory of $\mathbf{PER}(D)$ on the admissible per's.

The following theorem is well known [2] but we recall the proof for the readers benefit.

Theorem 2.5. *The category* $\mathbf{AP}(D)$ *is a sub-cartesian closed category of* $\mathbf{PER}(D)$.

Proof. We recall the constructions. It is straightforward to verify that the resulting per's are admissible. The terminal object 1 is the admissible per defined by

$$d \; 1 \; d' \Leftrightarrow d = \bot_D = d'.$$

The binary product of R and S is

$$\begin{gathered} d \; R \times S \; d' \\ \Updownarrow \\ \exists d_1, d_2, d_1', d_2' \in D.\ d = \langle d_1, d_2 \rangle \quad \wedge \quad d' = \langle d_1', d_2' \rangle \quad \wedge \quad d_1 \; R \; d_1' \quad \wedge \quad d_2 \; S \; d_2' \end{gathered}$$

The exponential of R and S, S^R, is given by

$$d \; S^R \; d' \quad \Leftrightarrow \quad \Phi(d) \simeq_{R,S} \Phi(d').$$

□

Lemma 2.6. *There is a faithful functor Classes*: $\mathbf{AP}(D) \to \mathbf{Set}$ *mapping an admissible per to the set of equivalence classes and an equivalence class of realizers to the map of equivalence classes they induce. This functor preserves products, i.e., for any pair of admissible per's* R, S, *Classes*$(R \times S) \cong$ *Classes*$(R) \times$ *Classes*(S).

Proof. *Classes* is the global sections functor, $hom(1, -)$, which preserves products. That it is faithful follows from the fact that all constant functions $D \to D$ are continuous. □

Definition 2.7. The category $\mathbf{AP}(D)_\bot$ of admissible per's and strict continuous functions is the full-on-objects subcategory of $\mathbf{AP}(D)$ with morphisms $[\alpha] \colon R \to S$ equivalence classes in $\mathcal{F}(R, S)_\bot / \simeq_{R,S}$.

Remark 2.8. Note that in $\mathbf{AP}(D)_\bot$, morphisms are required to have a *strict* continuous realizer. On the other hand, if there is a realizer $\alpha \in \mathcal{F}(R, S)$ with $\alpha(\bot_D) \; S \; \bot_D$ then the function that maps $\bot_D$ to $\bot_D$ and all other $d \in D$ to $\alpha(d)$ will still be continuous and equivalent to α in $\mathcal{F}(R, S)$. This function will thus be a realizer in $\mathcal{F}(R, S)_\bot$.

Theorem 2.9. $\mathbf{AP}(D)_\bot$ *is a cartesian sub-category of* $\mathbf{AP}(D)$.

Proof. Obvious since π, π', and $\langle \cdot, \cdot \rangle$ are strict. □

Theorem 2.10. *The category* $\mathbf{AP}(D)_\bot$ *is symmetric monoidal closed.*

Proof. The tensor of R and S is

$$\begin{gathered} d \; R \otimes S \; d' \\ \Updownarrow \\ d \; R \times S \; d' \\ \vee \\ \left(\begin{array}{c} \exists d_1, d_1' \in |R|.\ \exists d_2, d_2' \in |S|.\ d = \langle d_1, d_2 \rangle \quad \wedge \quad d' = \langle d_1', d_2' \rangle \quad \wedge \\ (d_1 \; R \; \bot_D \quad \vee \quad d_2 \; S \; \bot_D) \quad \wedge \quad (d_1' \; R \; \bot_D \quad \vee \quad d_2' \; S \; \bot_D) \end{array} \right) \end{gathered}$$

This complicated looking definition is most easily understood through the functor *Classes*: The equivalence classes of the tensor product are those of the product with the modification that all pairs where one of the coordinates are related to $\bot_D$ have been gathered into one big equivalence class.

The unit of the tensor I is defined by

$$dId' \quad \Leftrightarrow \quad d = d' = \bot_D \vee d = d' = \mathbf{i}.$$

The exponential of R and S, $R \multimap S$, is given by

$$d \; R \multimap S \; d' \quad \Leftrightarrow \quad d \; S^R \; d' \; \wedge \; (d'' \; R \; \bot_D \Rightarrow \Phi(d)(d'') \; S \; \bot_D \; S \; \Phi(d')(d''))$$

The proof consist of a series of straightforward verifications. □

For later use we shall mention how regular subobjects look in this category. We use $A \rightarrowtail R$ to express that A is a regular subobject of R, if R is an admissible per.

Lemma 2.11. *There is a bijective correspondence between regular subobjects of R and per's A such that*

$$\textit{Classes}(A) \subseteq \textit{Classes}(R) \wedge A \in \mathrm{Obj}(\mathbf{AP}(D)_\bot)$$

Proof. Assume R and A with the mentioned properties. Define R_A by

$$\begin{gathered} d \; R_A \; d' \\ \Updownarrow \\ \begin{pmatrix} d = \langle d_b, \bot_D \rangle & \wedge \\ d' = \langle d'_b, \bot_D \rangle & \wedge \\ d_b \; R \; d'_b \end{pmatrix} \quad \vee \quad \begin{pmatrix} d = \langle d_1, \mathbf{i} \rangle & \wedge \\ d' = \langle d'_1, \mathbf{i} \rangle & \wedge \\ d_1 \; R \; d'_1 \end{pmatrix} \quad \vee \\ \begin{pmatrix} d = \langle d_b, \bot_D \rangle & \wedge \\ d' = \langle d'_1, \mathbf{i} \rangle & \wedge \\ d_b \; A \; d'_1 \end{pmatrix} \quad \vee \quad \begin{pmatrix} d = \langle d_1, \mathbf{i} \rangle & \wedge \\ d' = \langle d'_b, \bot_D \rangle & \wedge \\ d_1 \; A \; d'_b \end{pmatrix} \end{gathered}$$

i.e pairs from $R \times \{\{\bot_D\}, \{\mathbf{i}\}\}$ with the added relations of pairs with their first components related in A. $R_A \in \mathrm{Obj}(\mathbf{AP}(D)_\bot)$ and there are two morphisms $R \multimap R_A$ realized by $d \mapsto \langle d, \bot_D \rangle$ and $d \mapsto \langle d, \mathbf{i} \rangle$ respectively. In view of remark 2.8, and since $\bot_D \; A \; \bot_D$, the latter does in fact realize a morphism of $\mathbf{AP}(D)_\bot$ and A is the equalizer of these two morphisms.

Conversely, the image of an equalizer is easily seen to be admissible. Thus all regular subobjects have a representative, which is a subset of the equivalence classes as desired. □

We also need the following fact about admissible per's

Lemma 2.12. *If I is an arbitrary set, and for all $i \in I$, R_i is an admissible per over D then the relation $\bigcap_{i \in I} R_i$ defined as*

$$d \bigcap_{i \in I} R_i \; d' \Leftrightarrow \forall i \in I. \, d \; R_i \; d'$$

is an admissible per over D.

2.1 Lifting

We now define a lifting functor, to establish a left adjoint to the inclusion $U\colon \mathbf{AP}(D) \to \mathbf{AP}(D)_\perp$. Define the map $L_0\colon \operatorname{Obj}(\mathbf{AP}(D)) \to \operatorname{Obj}(\mathbf{AP}(D)_\perp)$ by

$$\begin{array}{ccc} d & L_0(R) & d' \\ & \Updownarrow & \\ d = \bot_D = d' & \vee & \exists e, e' \in D.\big(d = \langle \mathbf{i}, e\rangle \ \wedge\ d' = \langle \mathbf{i}, e'\rangle \ \wedge\ e \, R \, e'\big) \end{array}$$

This is well-defined as $L_0(R)$ easily is admissible if R is.

Notice the "if" construct available on a lifted per: For R an admissible per if d is in the domain of $L_0(R)$ then d is either $\bot_D$ or a pair $\langle \mathbf{i}, e\rangle$. Hence $\Phi(\pi(d))$ is either the totally undefined function or the identity on D. Thus $\Phi(\pi(d))(d')$ can be read "if $d \notin [\bot]_{L_0(R)}$ then d' else $\bot_D$", where $[\bot]_S$ is the class represented by $\bot_D$ in the admissible per S.

We also have a "lift" map $\eta\colon R \to L_0(R)$ realized by $\lambda d \in D.\langle \mathbf{i}, d\rangle$ and an "unlift" map $\epsilon\colon L_0(R) \to R$ realized by π'. Notice that ϵ is strict, but η is not.

To handle morphisms we work at the level of realizers. Define, for admissible per's R and S, the map $L_1'\colon \mathcal{F}(R, S) \to \mathcal{F}(L_0(R), L_0(S))_\perp$ by

$$L_1'(\alpha) = \lambda d \in D.\Phi(\pi(d))(\langle \mathbf{i}, \alpha(\pi'(d))\rangle)$$

which reads "if $d \notin [\bot]_{L_0(R)}$ then lift(α(unlift d)) else $\bot_D$". As $L_1'(\alpha)(\langle \mathbf{i}, e\rangle) = \langle \mathbf{i}, \alpha(e)\rangle$, this is easily seen to be well-defined. As it also takes equivalent realizers to equivalent realizers, we can lift the map to the level of morphisms and a straightforward verification shows that this together with L_0 defines a functor $L\colon \mathbf{AP}(D) \to \mathbf{AP}(D)_\perp$.

Theorem 2.13. *There is a monoidal adjunction $L \dashv U$.*

Proof. One first shows that L is left adjoint to U in the ordinary sense. The unit of the adjunction is given by $(\eta_R\colon R \to UL(R))_{R \in APD_0}$, and for $t\colon R \to U(S)$ in $\mathbf{AP}(D)_\perp$, the required unique $u\colon L(R) \to S$ in $\mathbf{AP}(D)_\perp$, such that $U(u) \circ \eta_R = t$, is given by the realizer

$$\alpha_u = \lambda d \in D.\text{if } d \notin [\bot]_{L(R)} \text{ then } \alpha_t(\text{unlift } d) \text{ else } [\bot]_S$$

where α_t is a realizer for t.

To show that the adjunction is monoidal it suffices by [10] to show that the left adjoint L is a strong symmetric monoidal functor (see [16] for an explanation). To this end, we must exhibit an isomorphism $m_I\colon I \to L(1)$ and a natural isomorphism $m_{R,S}\colon L(R) \otimes L(S) \to L(R \times S)$. This is mostly straightforward; we just include the definition of $m_{R,S}$: it is the morphism realized by

```
$\lambda d \in D.$
    if $\pi(d) \neq \bot$ then
        if $\pi'(d) \neq \bot$ then
            lift of $\langle$unlift($\pi(d)$), unlift($\pi'(d)$)$\rangle$
        else $\bot_D$
    else $\bot_D$.
```

The inverse is realized by

```
$\lambda d \in D.$
    if $d \neq \bot$ then
        $\langle$lift of $\pi$(unlift($d$)), lift of $\pi'$(unlift($d$))$\rangle$
    else $\bot_D$.
```

□

2.2 Relation to axiomatic domain theory

We have advertised the slogan, that the theory of admissible per's is "a domain theory for polymorphism". In this section we explore different categories of admissible and chain complete per's and their relations, and relate the results to classical domain theory. The reader should keep the following picture in mind from classical domain theory.

$$\begin{tikzcd} \mathbf{Cpo} \arrow[r, bend right, "U"'] \arrow[r, phantom, "\top"] & \mathbf{pCpo} \arrow[l, bend right, "L"'] \arrow[r, "\cong"] & \mathbf{Cppo}_\perp \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] & \mathbf{Cppo} \arrow[l, bend right, "L"'] \end{tikzcd} \tag{1}$$

Here $\mathbf{Cpo}$ is the category of complete partial orders (cpo's), $\mathbf{pCpo}$ of cpo's and continuous partial functions, $\mathbf{Cppo}_\perp$ of pointed cpo's and strict continuous functions and $\mathbf{Cppo}$ of pointed cpo's and all continuous functions. In the diagram U always denotes inclusion and L lifting.

In axiomatic domain theory much focus has been on the leftmost adjunction, as in Fiore's thesis in which categories of partial maps are studied. The category of partial maps $\mathbf{pCpo}$ is isomorphic to the Kleisli category for the lifting monad on $\mathbf{Cpo}$ induced by the adjunction $U \dashv L$, and this is also isomorphic to the Eilenberg-Moore category for the monad and to $\mathbf{Cppo}_\perp$.

In PILL$_Y$, the adjunction on the right is axiomatized, and in general PILL$_Y$-models there is a priori no category corresponding to $\mathbf{Cpo}$. In the theory of admissible per's, however, there is one such, namely the category $\mathbf{CCP}(D)$ of chain complete per's over D with maps defined as in $\mathbf{AP}(D)$. One may easily show that the lifting functor of Section 2.1 extends to a functor $L\colon \mathbf{CCP}(D) \to \mathbf{AP}(D)_\perp$, and in fact this is left adjoint to the inclusion $U\colon \mathbf{AP}(D)_\perp \to \mathbf{CCP}(D)$, thus UL induces a monad on $\mathbf{CCP}(D)$. The picture corresponding to (1) for admissible per's is

$$\begin{tikzcd} \mathbf{CCP}(D) \arrow[r, bend right] \arrow[r, phantom, "\top"] \arrow[dr, bend right] \arrow[dr, phantom, "\top"] & \mathbf{AP}(D)_\perp \arrow[l, bend right] \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] & \mathbf{AP}(D) \arrow[l, bend right, "L"'] \, . \\ & \mathbf{CCP}(D)_{UL} \arrow[ul, bend right] \arrow[u, hook] & \end{tikzcd}$$

Here $\mathbf{CCP}(D)_{UL}$ is the Kleisli category for the monad. We will show that $\mathbf{AP}(D)_\perp$ is the Eilenberg-Moore category of the monad on $\mathbf{CCP}(D)$, but that this is not the same as $\mathbf{CCP}(D)_{UL}$ in the sense that the comparison map, which is the inclusion in the diagram, is not an isomorphism, as is the situation in domain theory.

Proposition 2.14. *The category $\mathbf{AP}(D)_\perp$ is equivalent to the Eilenberg-Moore category for UL on $\mathbf{CCP}(D)$.*

Proof. A standard theorem of adjunctions tells us that $\mathbf{AP}(D)_\perp$ is included in the Eilenberg Moore category. In fact, the inclusion maps an object R of $\mathbf{AP}(D)_\perp$ to the counit of the adjunction at R. We must show that any monad algebra for UL is of this form (up to isomorphism). Suppose $f\colon LS \to S$ is an algebra realized by α. Construct the admissible per S' by adding $\perp$ to the equivalence class of $\alpha(\perp)$ in S. It is now an easy check to show that $f\colon LS \to S$ is isomorphic as an algebra to the counit $\epsilon\colon LS' \to S'$. □

We remark that in fact, $\mathbf{CCP}(D)$ is a cartesian closed category, UL a strong commutative monad, and the symmetric monoidal structure on $\mathbf{AP}(D)_\perp$ is induced by UL as in [12].

Proposition 2.15. *The Kleisli category for the monad UL on $\mathbf{CCP}(D)$ is equivalent to the full subcategory of $\mathbf{AP}(D)_\perp$ on per's R such that $[\perp]_R = \{\perp\}$.*

Proof. The Kleisli category is isomorphic to the category of free algebras, which is equivalent to the mentioned category. □

As mentioned, this is different from the situation in classical domain theory, where the Kleisli category for the lifting monad on $\mathbf{Cpo}$ coincide with the Eilenberg-Moore category for the same monad, and both are isomorphic to $\mathbf{Cppo}_\perp$. For a simple example of an algebra for UL that is not isomorphic to a free one, suppose $\perp \neq d < e$ are elements of D, and consider the admissible per given by the collection of equivalence classes $\{\{\perp, e\}, \{d\}\}$.

The last proposition of this section shows how to recover $\mathbf{CCP}(D)$ from $\mathbf{AP}(D)_\perp$. This is interesting, as PILL$_Y$ is meant to axiomatize the adjunction to the right of (1), and in a general PILL$_Y$-model there is a priori no category corresponding to $\mathbf{Cpo}$.

Proposition 2.16. *The co-Eilenberg-Moore category for the comonad LU on $\mathbf{AP}(D)_\perp$ is equivalent to $\mathbf{CCP}(D)$.*

Proof. We show that the co-Eilenberg-Moore category is isomorphic to the category of admissible per's R for which the equivalence class $[\perp_D]$ is a downward closed subset of the domain of R — i.e., if $d\ R\ d$, $d \leq d'$ and $d'\ R\ \perp_D$, then $d\ R\ \perp_D$ — and maps that preserve and reflect $[\perp_D]$. This category is equivalent to $\mathbf{CCP}(D)$, with one map of the equivalence lifting a chain complete per, and the other discarding the equivalence class $[\perp_D]$ from an admissible per.

Suppose α is a realizer for a coalgebra ξ on an admissible per R, and $d\ R\ d$, $d \leq d'$ and $d'\ R\ \perp$. Since α is strict, $\alpha(\perp) = \perp$, and so $\alpha(d')\ LUR\ \perp$ implying $\alpha(d') = \perp$. Thus, by monotonicity $\alpha d = \perp$. Since $\epsilon \circ \xi$ is the identity, where ϵ is the counit, $d\ R\ \perp$. On the other hand, if $[\perp]$ is a downward closed subset of the domain of R then one may easily check that

$$\xi(d) = \begin{cases} \perp & \text{if } \exists d' \geq d.\, d'\ R\ \perp \\ \langle \mathbf{i}, d\rangle & \text{else} \end{cases}$$

defines a unique coalgebra structure on R. Continuity of ξ follows from admissibility of R.

Suppose $t \colon R \multimap S$ is a map between such per's, preserving coalgebra structure. Since t has a strict realizer it must preserve the equivalence class of $\perp$. To see that it also reflects it, suppose $t([d]_R) = [\perp]_S$. Then also $LU(t)(\xi_R([d])) = [\perp]_{LUS}$ implying that $\xi_R([d]) = [\perp]_{LR}$. Clearly, then $d\ R\ \perp$.

Suppose on the other hand that $t \colon R \multimap S$ reflects the equivalence class of $\perp$. In order to show $LU(t) \circ \xi_R = \xi_S \circ t$ we write them out, assuming t is realized by α_t:

$$LU(t)(\xi_R([d])) = \begin{cases} LU(t)([\perp]) & \text{if } \exists d' \geq d.\, d'\ R\ \perp \\ LU(t)([\langle \mathbf{i}, d\rangle]) & \text{else} \end{cases} = \begin{cases} [\perp] & \text{if } \exists d' \geq d.\, d'\ R\ \perp \\ [\langle \mathbf{i}, \alpha_t d\rangle] & \text{else} \end{cases}$$

and

$$\xi_S(t([d])) = \begin{cases} [\perp] & \text{if } \exists d' \geq \alpha_t d.\, d'\ S\ \perp \\ [\langle \mathbf{i}, \alpha_t d\rangle] & \text{else} \end{cases}$$

Using that for $d\ R\ d$, $\exists d' \geq \alpha_t d.\, d'\ S\ \perp \Leftrightarrow d\ R\ \perp$, we can rewrite them to

$$LU(t)(\xi_R([d])) = \begin{cases} [\perp] & \text{if } d\ R\ \perp \\ [\langle \mathbf{i}, \alpha_t d\rangle] & \text{else} \end{cases} \quad \text{and} \quad \xi_S(t([d])) = \begin{cases} [\perp] & \text{if } \alpha_t d\ S\ \perp \\ [\langle \mathbf{i}, \alpha_t d\rangle] & \text{else} \end{cases}$$

which are equal since t reflects $[\perp]$. □

3 A domain-theoretic PILL$_Y$ model

The calculus PILL$_Y$ is a Polymorphic Intuitionistic / Linear Lambda calculus with a fixed point combinator Y. It is basically DILL of [3] extended with polymorphism and fixed points. Types are formed using the grammar

$$\sigma ::= \alpha \mid I \mid \sigma \otimes \tau \mid \sigma \multimap \tau \mid !\sigma \mid \prod \alpha.\, \sigma.$$

Terms are written in context as

$$\Xi \mid \Gamma; \Delta \vdash t \colon \sigma$$

where Ξ is the context of free type variables, Γ is a context of inituitionistic variables and Δ is a context of linear variables. All the free type variables occurring in Γ, Δ and σ must be in Ξ. The typing rules for terms are presented in Figure 1.

The type constructor $\multimap$ denotes a linear function space, and its constructor is a lambda abstraction for linear variables. Intuitionistic function space can be encoded using the Girard encoding $\sigma \to \tau = !\sigma \multimap \tau$. Using this encoding, the polymorphic fixed point combinator Y has the type $\prod \alpha.\, (\alpha \to \alpha) \to \alpha$.

Terms of PILL$_Y$ are considered up to an equality theory including standard β, η rules and stating that Y is a fixed point operator. For further details on PILL$_Y$ see [6].

This section presents a PILL$_Y$ model in which the $\otimes$ and $\multimap$ are interpreted using the symmetric monoidal closed structure on $\mathbf{AP}(D)_\perp$, and ! is interpreted using lifting. But because PILL$_Y$ contains polymorphism the categorical formulation of the model structure is based on fibred category theory. A model of PILL$_Y$ is essentially a fibred model of DILL [3] with extra structure to model polymorphism.

The model to be constructed here will be denoted

$$\begin{tikzcd}
\mathbf{UFam}(\mathbf{AP}(D)_\perp) \arrow[rr, bend right, "U"'] \arrow[rr, phantom, "\perp"] \arrow[dr, "q"'] & & \mathbf{UFam}(\mathbf{AP}(D)) \arrow[ll, bend right, "L"'] \arrow[dl, "p"] \\
& \mathbf{Set}. &
\end{tikzcd} \tag{2}$$

The fibred adjunction of (2) is a fibred version of the adjunction between $\mathbf{AP}(D)$ and $\mathbf{AP}(D)_\perp$. The calculus PILL$_Y$ will be modeled in the fibration q using the symmetric monoidal closed structure to model the type constructions $I, \otimes, \multimap$. The lifting functor L will be used to model ! and polymorphism will be modeled via simple products with respect to a generic object. A term $\vec{x}\colon \vec{\sigma}; \vec{y}\colon \vec{\sigma}' \vdash t\colon \tau$ in which the x_i are the intuitionistic variables and the y_j are the linear variables is modeled as a vertical morphism $\bigotimes_i LU[\![\sigma_i]\!] \otimes \bigotimes_j [\![\sigma'_j]\!] \to [\![\tau]\!]$ in the fibration q. The fibration p still plays a role as it can be used to model the terms with only intuitionistic variables.

We shall only show that the categorical structure needed for modeling PILL$_Y$ is present, and not spell out the interpretation of PILL$_Y$ in the model. For further details on PILL$_Y$ models see [16].

Define the contravariant functor $P : \mathbf{Set}^{\mathrm{op}} \to \mathbf{Cat}$ by mapping a set I to the category $P(I)$ with

Objects: $(R_i)_{i\in I}$ where for all $i \in I$, R_i is an object of $\mathbf{AP}(D)$.

Morphisms: $(t_i)_{i\in I}\colon (R_i)_{i\in I} \to (S_i)_{i\in I}$, where, for all $i \in I$, $t_i \in \mathbf{AP}(D)(R_i, S_i)$ and the t_i have a *uniform realizer* in the sense that there exists an α in $[D \to D]$ such that for all $i \in I$, $t_i = [\alpha]_{\simeq_{R_i,S_i}}$.

$$\frac{}{\Xi \mid \Gamma; - \vdash \star \colon I} \qquad \frac{}{\Xi \mid \Gamma; - \vdash Y \colon \prod \alpha. !(!\alpha \multimap \alpha) \multimap \alpha}$$

$$\frac{}{\Xi \mid \Gamma, x \colon \sigma; - \vdash x \colon \sigma} \qquad \frac{}{\Xi \mid \Gamma; x \colon \sigma \vdash x \colon \sigma}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash t \colon \sigma \multimap \tau \quad \Xi \mid \Gamma; \Delta' \vdash u \colon \sigma}{\Xi \mid \Gamma; \Delta, \Delta' \vdash t\, u \colon \tau}\ \Delta, \Delta' \text{ disjoint}$$

$$\frac{\Xi \mid \Gamma; \Delta, x \colon \sigma \vdash u \colon \tau}{\Xi \mid \Gamma; \Delta \vdash \lambda^{\circ} x \colon \sigma. u \colon \sigma \multimap \tau}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash t \colon \sigma \quad \Xi \mid \Gamma; \Delta' \vdash s \colon \tau}{\Xi \mid \Gamma; \Delta, \Delta' \vdash t \otimes s \colon \sigma \otimes \tau}\ \Delta, \Delta' \text{ disjoint}$$

$$\frac{\Xi \mid \Gamma; - \vdash t \colon \sigma}{\Xi \mid \Gamma; - \vdash !t \colon \sigma}$$

$$\frac{\Xi, \alpha \colon \mathsf{Type} \mid \Gamma; \Delta \vdash t \colon \sigma}{\Xi \mid \Gamma; \Delta \vdash \Lambda \alpha \colon \mathsf{Type}. t \colon \prod \alpha \colon \mathsf{Type}. \sigma}\ \Xi \mid \Gamma; \Delta \text{ is well-formed}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash t \colon \prod \alpha \colon \mathsf{Type}. \sigma \qquad \Xi \vdash \tau \colon \mathsf{Type}}{\Xi \mid \Gamma; \Delta \vdash t(\tau) \colon \sigma[\tau/\alpha]}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash s \colon \sigma \otimes \sigma' \qquad \Xi \mid \Gamma; \Delta', x \colon \sigma, y \colon \sigma' \vdash t \colon \tau}{\Xi \mid \Gamma; \Delta, \Delta' \vdash \text{let } x \colon \sigma \otimes y \colon \sigma' \text{ be } s \text{ in } t \colon \tau}\ \Delta, \Delta' \text{ disjoint}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash s \colon !\sigma \qquad \Xi \mid \Gamma, x \colon \sigma; \Delta' \vdash t \colon \tau}{\Xi \mid \Gamma; \Delta, \Delta' \vdash \text{let } !x \colon !\sigma \text{ be } s \text{ in } t \colon \tau}\ \Delta, \Delta' \text{ disjoint}$$

$$\frac{\Xi \mid \Gamma; \Delta \vdash t \colon I \quad \Xi \mid \Gamma; \Delta' \vdash s \colon \sigma}{\Xi \mid \Gamma; \Delta, \Delta' \vdash \text{let } \star \text{ be } t \text{ in } s \colon \sigma}$$

Figure 1: Typing rules for PILL$_Y$ terms

For a function $f\colon I \to J$, the reindexing functor $P(f)$ is simply given by composition with f.

Define the contravariant functor $Q : \mathbf{Set}^{\mathrm{op}} \to \mathbf{Cat}$ given by mapping set I to the category $Q(I)$ with

Objects: $(R_i)_{i\in I}$ where for all $i \in I$, R_i is an object of $\mathbf{AP}(D)_\perp$.

Morphisms: $(t_i)_{i\in I}\colon (R_i)_{i\in I} \to (S_i)_{i\in I}$ where for all $i \in I$, $t_i \in \mathbf{AP}(D)_\perp(R_i, S_i)$ and $\exists \alpha \in [D \to D].\, \forall i \in I.\, t_i = [\alpha]_{\simeq_{R_i,S_i}}$.

For a function $f\colon I \to J$, the reindexing functor $Q(f)$ is again simply given by composition with f.

That we have two contravariant functors is obvious. The Grothendieck construction (see for example [13]) then gives us two split fibrations, $p\colon \mathbf{UFam}(\mathbf{AP}(D)) \to \mathbf{Set}$ and $q\colon \mathbf{UFam}(\mathbf{AP}(D)_\perp) \to \mathbf{Set}$. The functors L and U both operate one the level of realizers and so lift to fibred functors between these two fibrations (we abuse notation and also denote the fibred functors by L and U). Explicitly, on objects $L(I,(R_i)_{i\in I}) = (I,(L(R_i))_{i\in I})$ and on vertical morphisms $L(I,(t_i)_{i\in I}) = (I,(L(t_i))_{i\in I})$. Likewise for U. These are not recursive definitions, they simply look so because of the reuse of letters.

Proposition 3.1. *L and U are split fibred functors and $L \dashv U$ is a split fibred strong monoidal adjunction*

Proof. It is obvious that L and U are split fibred functors; the second part follows immediately from Theorem 2.13. ☐

To show that (2) is a model of PILL it remains to be shown that q has a generic object and simple products, in other words models polymorphism.

Lemma 3.2. *The set $\Omega = \mathrm{Obj}(\mathbf{AP}(D)_\perp) = \mathrm{Obj}(\mathbf{AP}(D))$ is a split generic object of the fibration q. The fibration q has simple split Ω-products satisfying the Beck-Chevalley condition.*

Proof. The first part is obvious. For the second part, one uses the usual definition for uniform families of ordinary per's and verifies that it restricts to admissible per's: We recall from [13] that given any projection $\pi_M\colon M \times \Omega \to M$ in $\mathbf{Set}$, the right adjoint $\forall_M$ to π_M^* is given on objects by intersection:

$$\forall_M((R_{(a,\omega)})_{(a,\omega)\in M\times\Omega}) = (\bigcap_{\omega\in\Omega} R_{(a,\omega)})_{a\in M}.$$

By lemma 2.12 the resulting per is admissible. ☐

Theorem 3.3. *The diagram* (2) *constitutes a model of PILL$_Y$.*

Proof. Given the preceding results it only remains to verify that (1) the structure in the diagram models the polymorphic fixed point combinator and that (2) $\mathbf{UFam}(\mathbf{AP}(D))$ is equivalent to the category of products of free coalgebras of $\mathbf{UFam}(\mathbf{AP}(D))_\perp$.

For (1), the required follows, as expected, because the per's are strict and complete. In more detail, what is needed is an element of the PILL$_Y$ type $\prod \alpha.\, (\alpha \to \alpha) \to \alpha$ as interpreted in the model, giving fixed points to maps. An inspection of the model shows that this means a continuous function *Fix*: $[D \to D] \to D$ such that for any admissible per R, if α is a realizer for a map $L(R) \multimap R$, then *Fix*(α) is a fixed point for $\alpha \circ l$, where l is a realizer for the "lifting map" $\eta\colon R \to L(R)$ described in Section 2.1. Moreover, if $\alpha \simeq_{LR,R} \alpha'$ are related in $L(R) \multimap R$ then we must have $(\mathit{Fix}(\alpha), \mathit{Fix}(\alpha')) \in R$. Taking *Fix* to be the function $\alpha \mapsto \bigsqcup_n (\alpha \circ l)^n(\perp)$ gives an element clearly satisfying the first condition. The second condition is satisfied because R is strict and chain complete.

For (2), observe that by [16, Proposition 1.21] applied to Theorem 2.9 it suffices to show that $\mathbf{UFam}(\mathbf{AP}(D))$ is equivalent to the co-Kleisli category of the adjunction $L \dashv U$, but this follows from the fact that U is an inclusion surjective on objects. ☐

3.1 A parametric domain-theoretic model of $\mathbf{PILL}_Y$

In this section, we introduce a parametric version of the thus far constructed model. It is essentially obtained through a parametric completion process [25]. In [14] it is shown how the parametric completion process can be used to construct parametric LAPL-structures in general.

One of the reasons why having a parametric model is interesting, is that it will be a model of recursive types, containing solutions to recursive domain equations. Section 5 details the family of recursive domain equations, that can be solved in the obtained model.

We will arrive at the diagram

$$\begin{tikzcd}
\mathbf{PFam}(\mathbf{AP}(D)_\perp) \arrow[rr, bend right, "U"'] \arrow[rr, phantom, "\perp"] \arrow[dr] & & \mathbf{PFam}(\mathbf{AP}(D)) \arrow[ll, bend right, "L"'] \arrow[dl] \\
& \mathbf{PAP}(D). &
\end{tikzcd} \tag{3}$$

As is usual in the parametric completion process, types will be pairs (f^p, f^r) where f^p is a type in the sense of the model (2), and f^r is a relational interpretation of the type, i.e., a map taking a vector of relations and producing a new relation. In this setup, by relation on a pair of admissible per's R, S we shall mean a regular subobject of the product per $R \times S$ in $\mathbf{AP}(D)_\perp$. Since *Classes*$(R \times S) \cong$ *Classes*$(R) \times$ *Classes*(S), Lemma 2.11 gives the following characterization of the relations in question: these are subsets $M \subseteq$ *Classes*$(R) \times$ *Classes*(S) such that $([\perp]_R, [\perp]_S) \in M$, and if $(d_n), (d'_n)$ are increasing chains of elements of D in the domain of R and S respectively, such that $([d_n]_R, [d'_n]_S) \in M$ for all n, then also $([\bigsqcup_{n\in\omega} d_n]_R, [\bigsqcup_{n\in\omega} d'_n]_S) \in M$. (It is crucial that subobject is in the category with *strict* maps — this is what gives $([\perp]_R, [\perp]_S) \in M$). As always we write $A \rightarrowtail R \times S$ for such relations. We adopt the notation $\mathrm{RegSub}(R \times S)$ for the set of objects A in $\mathbf{AP}(D)_\perp$ such that $A \rightarrowtail R \times S$.

We now return to the definition of the fibrations of (3). The base category $\mathbf{PAP}(D)$ is defined as

Objects: $n \in N$ — objects are natural numbers.

Morphisms: $f \colon n \to m$ is an m-tuple, $(f_1, \ldots, f_m)$, where each f_i is a pair (f_i^p, f_i^r) satisfying

- f_i^p is a map of objects $(\mathrm{Obj}(\mathbf{AP}(D)_\perp))^n \to \mathrm{Obj}(\mathbf{AP}(D)_\perp)$
- f_i^r is a map, that to two n-tuples of objects of $\mathbf{AP}(D)_\perp$ associates a set-theoretic map of subobjects

  $$f_i^r \in \Pi_{\vec{R},\vec{S}\in(\mathrm{Obj}(\mathbf{AP}(D)_\perp))^n} \Big(\Pi_{j\in\{1,\ldots,n\}} \mathrm{RegSub}(R_j \times S_j) \to \mathrm{RegSub}(f_i^p(\vec{R}) \times f_i^p(\vec{S}))\Big)$$

  satisfying

  $$\forall \vec{R} \in (\mathrm{Obj}(\mathbf{AP}(D)_\perp))^n . f_i^r(\vec{R}, \vec{R})(\vec{eq_{R_j}}) = eq_{f_i^p(\vec{R})},$$

We now describe $\mathbf{PFam}(\mathbf{AP}(D)_\perp) \to \mathbf{PAP}(D)$ and $\mathbf{PFam}(\mathbf{AP}(D)) \to \mathbf{PAP}(D)$.

We plan to use the Grothendieck construction, and so define indexed categories: $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$ is defined with

Objects: morphisms in $\mathbf{PAP}(D)$ from n to 1.

Morphisms: $t \colon f \to g$ is a family of morphisms $(t_{\vec{R}} \colon f^p(\vec{R}) \to g^p(\vec{R}))_{\vec{R}\in(\mathrm{Obj}(\mathbf{AP}(D)_\perp))^n}$ of $\mathbf{AP}(D)_\perp$ with a uniform realizer (as in the definition of $\mathbf{UFam}(\mathbf{AP}(D))$) which respects relations in the sense that

$$\forall \vec{A} \rightarrowtail \vec{R} \times \vec{S} . f^r(\vec{R}, \vec{S}, \vec{A})([d], [d']) \Rightarrow g^r(\vec{R}, \vec{S}, \vec{A})(t_{\vec{R}}([d]), t_{\vec{S}}([d'])).$$

If we write $\mathbf{LR}(\mathbf{AP}(D)_\perp)_0$ for the collection of all admissible relations on admissible per's, and $(\mathbf{AP}(D)_\perp)_0$ for the collection of all admissible per's, then there is a reflexive graph

$$\begin{tikzcd}
(\mathbf{AP}(D)_\perp)_0 \arrow[r, shift left=1.5] & \mathbf{LR}(\mathbf{AP}(D)_\perp)_0 \arrow[l, shift right=3] \arrow[l, shift left=0]
\end{tikzcd}$$

where the two maps going left map a relation to its domain and codomain respectively and the map going right maps an admissible per to the equality relation. By this being a reflexive graph, we mean that going right and then back using either of the two maps is the identity. Another way to think of an object of $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$ is as a pair (f^r, f^p) in a diagram

$$\begin{tikzcd}
\mathbf{LR}(\mathbf{AP}(D)_\perp)_0^n \arrow[r, "f^r"] \arrow[d, shift right=2] \arrow[d, shift left=2] & \mathbf{LR}(\mathbf{AP}(D)_\perp)_0 \arrow[d, shift right=2] \arrow[d, shift left=2] \\
(\mathbf{AP}(D)_\perp)_0^n \arrow[r, "f^p"] \arrow[u] & (\mathbf{AP}(D)_\perp)_0 \arrow[u]
\end{tikzcd}$$

In the diagram the three obvious squares are required to commute. For example, the two ways of starting in the lower left corner and ending in the upper right are equal, which is exactly the requirement that f^r preserves equality.

Quite similarly $(\mathbf{PFam}(\mathbf{AP}(D)))_n$ is defined as the category with

Objects: morphisms in $\mathbf{PAP}(D)$ from n to 1.

Morphisms: $t\colon f \to g$ is a uniformly realized family of morphisms $(t_{\vec{R}})_{\vec{R}\in(\mathrm{Obj}(\mathbf{AP}(D)_\perp))^n}$ of $\mathbf{AP}(D)$ such that

$$t_{\vec{R}}\colon U(f^p(\vec{R})) \to U(g^p(\vec{R}))$$

where $U\colon \mathbf{AP}(D)_\perp \to \mathbf{AP}(D)$ is the forgetful functor. That we now ask for morphisms of $\mathbf{AP}(D)$ removes the demand, that the uniform realizer be strict. Again this t should respect relations:

$$\forall \vec{A} \rightarrowtail \vec{R}\times\vec{S}. f^r(\vec{R},\vec{S},\vec{A})([d],[d']) \Rightarrow g^r(\vec{R},\vec{S},\vec{A})(t_{\vec{R}}([d]), t_{\vec{S}}([d'])).$$

Note that the only difference between the two definitions is the choice of category in which the $t_{\vec{R}}$ are required to be morphisms.

We will very often write simply $f^r(\vec{A})$ for $f^r(\vec{R},\vec{S},\vec{A})$.

Definition 3.4. Define the functor $\mathbb{L}\colon \mathbf{PFam}(\mathbf{AP}(D)) \to \mathbf{PFam}(\mathbf{AP}(D)_\perp)$ on

objects by

$$\mathbb{L}((f^p, f^r)) = (F^p, F^r)$$

where

$$F^p(\vec{R}) = L(f^p(\vec{R}))$$

and

$$F^r((\vec{R},\vec{S},\vec{A})) = L(f^r(\vec{R},\vec{S},\vec{A}))$$

morphisms by

$$\mathbb{L}(t\colon (f^p, f^r) \to (g^p, g^r))(R) = L(t(R))$$

In the definition, we have lifted a relation. By this we mean to apply the lifting functor to the span $(\pi\circ f^r, \pi'\circ f^r)$ corresponding to the relation. The resulting relation relates lifted elements to each other iff the unlifted versions are related, and relates the equivalence classes of $\perp$ to each other. We define $\mathbb{U}\colon \mathbf{PFam}(\mathbf{AP}(D)_\perp) \to \mathbf{PFam}(\mathbf{AP}(D))$ in a similar way using U instead of L.

Lemma 3.5. $\mathbb{L}\colon \mathbf{PFam}(\mathbf{AP}(D)) \to \mathbf{PFam}(\mathbf{AP}(D)_\perp)$ *and* $\mathbb{U}\colon \mathbf{PFam}(\mathbf{AP}(D)_\perp) \to \mathbf{PFam}(\mathbf{AP}(D))$ *are both fibred functors, and constitute a fibred adjunction* $\mathbb{L} \dashv \mathbb{U}$*.*

By an easy extension of Theorem 2.5, we have:

Proposition 3.6. $\mathbf{PFam}(\mathbf{AP}(D))$ *is fibred cartesian closed.*

Proof. The product of (f^p, f^r) and (g^p, g^r) is $(f^p \times g^p, f^r \times g^r)$ where $f^p \times g^p$ is the point-wise product, and $f^r \times g^r$ takes the point-wise product of subobjects, which of course is a subobject of the products. In the exponent $(f^p \to g^p, f^r \to g^r)$ the first component is defined point wise, and the second component $f^r \to g^r$ relates the equivalence classes $[d], [d']$ if they map related elements to related elements in the sense that if $([e], [e']) \in f^r(\vec{A})$ then $([\Phi(d)e], [\Phi(d')e']) \in g^r(\vec{A})$. (Recall that the latter is well defined, i.e., independent of the choice of representatives). □

Proposition 3.7. $\mathbf{PFam}(\mathbf{AP}(D)_\perp)$ *is fibred cartesian and fibred symmetric monoidal closed.*

Proof. We just present the SMCC structure: In the fibre $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$, the tensor product of (f^p, f^r) and (g^p, g^r) is $(f^p \otimes g^p, f^r \otimes g^r)$ where $(f^p \otimes g^p)(\vec{R}) = f^p(\vec{R}) \otimes g^p(\vec{R})$ and $f^r(\vec{A}) \otimes g^r(\vec{A})$ is the image of $f^r(\vec{A}) \times g^r(\vec{A})$ under the quotient map from the product to the tensor. In other words $f^r(\vec{A}) \otimes g^r(\vec{A})$ relates the equivalence classes of $\perp$ and relates $[\langle d, d' \rangle]_{f^p(\vec{R}) \otimes g^p(\vec{R})}$ to $[\langle e, e' \rangle]_{f^p(\vec{S}) \otimes g^p(\vec{S})}$ (assuming these are not representatives of the $[\perp]$ equivalence classes) if ([
$elAl], [e]) \in f^r(\vec{A})$ and $([d'], [e']) \in g^r(\vec{A})$.

The unit of the tensor is given by the object $(\vec{R} \mapsto I, \vec{A} \mapsto eq_I)$.

The exponential of (f^p, f^r) and (g^p, g^r) in $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$, is $(f^p \multimap g^p, f^r \multimap g^r)$ where again $f^p \multimap g^p$ is defined pointwise using the closed structure of $\mathbf{AP}(D)_\perp$, i.e., $(f^p \multimap g^p)(\vec{R}) = f^p(\vec{R}) \multimap g^p(\vec{R})$. The relational interpretation of the exponential $(f^r \multimap g^r)(\vec{A})$ relates equivalence classes that represent maps that preserve relations, i.e., $(f^r \multimap g^r)(\vec{A})([d], [d'])$ iff

$$\forall([e], [e']) \in f^r(\vec{A}).\, ([\Phi(d)(e)], [\Phi(d')(e')]) \in g^r(\vec{A}).$$

To verify the adjunction $(-) \otimes (f^p, f^r) \dashv (f^p, f^r) \multimap (-)$, we use that we know it holds in the first component and then check that the bijection can be restricted to realizers that define morphisms in the second component; the latter is a direct consequence of the way the relational interpretations of $\otimes$ and $\multimap$ are defined. □

Lemma 3.8. $\mathbb{L} \dashv \mathbb{U}$ *is a fibred symmetric monoidal adjunction.*

Proof. This proceeds much as in the unfibred case. We show that $\mathbb{L}$ is a fibred strong symmetric monoidal functor. We must provide a morphism m_I and a natural transformation m, but we can simply use the same realizers as before, since everything has been defined coordinatewise and these realizers are independent of the specific per's, and hence are uniform realizers. □

The next lemma shows that (3) models polymorphism.

Lemma 3.9. *The fibration* $\mathbf{PFam}(\mathbf{AP}(D)_\perp) \to \mathbf{PAP}(D)$ *has a split generic object* Ω *and simple* Ω*-products.*

Proof. Clearly $\Omega = 1$ is a split generic object. For the simple products, given a projection $\pi\colon n+1 \to n$, we must define a right adjoint to π^*. The construction is exactly as in [13, Section 8.4]: the adjoint maps an object (f^p, f^r) of $\mathbf{PFam}(\mathbf{AP}(D)_\perp)_{n+1}$ to $(\prod f^p, \prod f^r)$ in $\mathbf{PFam}(\mathbf{AP}(D)_\perp)_n$, where

$$(\textstyle\prod f^p)(R_1, \ldots, R_n)(d, e)$$

iff

$$\forall R.\, f^p(R_1,\ldots,R_n,R)(d,e)\wedge$$
$$\forall R,S,A \rightarrowtail R\times S.\,([d]_{f^p(R_1,\ldots,R_n,R)},[e]_{f^p(R_1,\ldots,R_n,S)})\in f^r(eq_{R_1},\ldots,eq_{R_n},A)$$

and $\prod f^r(A_1,\ldots,A_n)([d],[e])$ iff

$$\forall R,S,A\rightarrowtail R\times S.\, f^r(A_1,\ldots,A_n,A)([d],[e]).$$

For this to be an object of $\mathbf{PFam}(\mathbf{AP}(D)_\perp)_n$ one needs to check that in fact $\prod f^r(eq_{R_1},\ldots,eq_{R_n})$ is equality on $\prod f^p(\vec{R})$, but this can easily be verified. $\square$

Proposition 3.10. *The diagram (3) constitutes a PILL$_Y$ model.*

Proof. It only remains to verify that the structure models the fixed point combinator. Here we simply use the Y from Theorem 3.3, which works since relations are strict and chain complete. $\square$

Remark 3.11. Notice that in the model (3), the fibre of closed types, i.e., the category $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_0$ is isomorphic to $\mathbf{AP}(D)_\perp$.

3.2 Overview of Interpretation

We can summarize the interpretation of types.

Recall, that the interpretation of a type $\alpha_1,\ldots,\alpha_n\vdash\sigma$ is a pair (f^p,f^r), where f^p is a function that takes n admissible per's (detailing the types for the free type variables) and produces an admissible per, and f^r is the relational interpretation. Thus f^r takes n regular subobjects $A_1\rightarrowtail R_1\times S_1,\ldots,A_n\rightarrowtail R_n\times S_n$ and gives a regular subobject of $f^p(R_1,\ldots,R_n)\times f^p(S_1,\ldots,S_n)$.

Assume $\alpha_1,\ldots,\alpha_n\vdash\sigma$, and that $R_1,\ldots,R_n$ and $S_1,\ldots,S_n$ are admissible per's and $A_1\rightarrowtail R_1\times S_1,\ldots,A_n\rightarrowtail R_n\times S_n$ are regular subobjects in $\mathbf{AP}(D)_\perp$. Then the interpretation of σ is given by the following two tables:

σ	$f^p(R_1,\ldots,R_n)$
α_i	R_i
I	$\{(\perp_D,\perp_D),(\mathbf{i},\mathbf{i})\}$
$\tau\otimes\tau'$	$\{(\langle d_1,d_2\rangle,\langle d_1',d_2'\rangle)\mid [\![\tau]\!]^p(d_1,d_1')\ \wedge\ [\![\tau']\!]^p(d_2,d_2')\}\quad\cup$ $\{(\langle d_1,d_2\rangle,\langle d_1',d_2'\rangle)\mid d_1,d_1'\in\lvert[\![\tau]\!]^p\rvert\wedge d_2,d_2'\in\lvert[\![\tau']\!]^p\rvert\wedge$ $([\![\tau]\!]^p(d_1,\perp_D)\vee[\![\tau']\!]^p(d_2,\perp_D))\wedge$ $([\![\tau]\!]^p(d_1',\perp_D)\vee[\![\tau']\!]^p(d_2',\perp_D))\}$
$\tau\multimap\tau'$	$\{(d,d')\mid\Phi(d)(\perp_D)=\Phi(d')(\perp_D)=\perp_D\wedge$ $\forall(e,e')\in[\![\tau]\!]^p.[\![\tau']\!]^p(\Phi(d)(e),\Phi(d')(e'))\}$
$!\tau$	$\{(\perp_D,\perp_D)\}\cup\{(\langle\mathbf{i},d\rangle,\langle\mathbf{i},d'\rangle)\mid[\![\tau]\!]^p(d,d')\}$
$\prod\alpha.\tau$	$\{(d,d')\mid\forall R\in\mathbf{AP}(D)_\perp.[\![\tau]\!]^p(R_1,\ldots,R_n,R)(d,d')\wedge$ $\forall R,S\in\mathbf{AP}(D)_\perp.\forall A\rightarrowtail R\times S.[\![\tau]\!]^r(eq_{R_1},\ldots,eq_{R_n},A)([d],[d'])\}$

σ	$f^r(A_1,\ldots,A_n)$
α_i	A_i
I	$\{([\bot_D],[\bot_D]),([\mathbf{i}],[\mathbf{i}])\}$
$\tau\otimes\tau'$	$\{([\langle d_1,d_2\rangle],[\langle d_1',d_2'\rangle]) \mid [\![\tau]\!]^r([d_1],[d_1']) \wedge [\![\tau']\!]^r([d_2],[d_2'])\}$
$\tau\multimap\tau'$	$\{([d],[d']) \mid \forall([e],[e'])\in[\![\tau]\!]^r.[\![\tau']\!]^r([\Phi(d)(e)],[\Phi(d')(e')])\}$
$!\tau$	$\{([\bot_D],[\bot_D])\}\cup\{([\langle\mathbf{i},d\rangle],[\langle\mathbf{i},d'\rangle]) \mid [\![\tau]\!]^r([d],[d'])\}$
$\prod\alpha.\tau$	$\{([d],[d']) \mid \forall R,S\in\mathbf{AP}(D)_\bot.\forall A\rightarrowtail R\times S.[\![\tau]\!]^r(A_1,\ldots,A_n,A)([d],[d'])\}$

4 A parametric LAPL-structure

Intuitively the PILL$_Y$ model constructed in Section 3.1 is parametric, because every type has a relational interpretation (f^r) satisfying identity extension (this is the requirement that $f^r(eq_{R_1},\ldots,eq_{R_n}) = eq_{f^p(\vec{R})}$), and moreover, the relational interpretations of $\multimap$ and $\prod$ are given by the usual interpretations as can be seen from the proofs above. In this section we make this statement precise by showing that the PILL$_Y$ model can be extended to a parametric LAPL-structure [5], i.e., a model of the logic for parametricity on PILL$_Y$ presented in [6]. This will give us proofs of encodings of recursive types in the model as we shall explain in Section 5 below.

The LAPL-structure will be given by the diagram

$$\begin{tikzcd}
 & & \mathbf{Fam}(\mathrm{Sub}(\mathbf{Set})) \arrow[d, "r"] \\
\mathbf{PFam}(\mathbf{AP}(D)_\bot) \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] \arrow[dr] & \mathbf{PFam}(\mathbf{AP}(D)) \arrow[l, bend right, "L"'] \arrow[r, hook, "I"] \arrow[d] & \mathbf{Fam}(\mathbf{Set}) \arrow[dl, "s"] \\
 & \mathbf{PAP}(D). &
\end{tikzcd} \quad (4)$$

The left hand side of the diagram is simply the model (3), which we want to reason about using the logic for parametricity. We use the logic of sets to reason about types in the model. We have already used the term admissible relation to refer to certain subsets of the product of sets of equivalence classes, and general propositions on admissible per's will be simply subsets of the set of equivalence classes for the per. Thus we include the category of admissible per's into the category of sets using the *Classes* functor, and reason using subsets. The inclusion of per's into the larger category of sets is needed because when reasoning about parametricity one needs to quantify over all relations between a pair of types, and the collection of relations between per's is a set, not a per. Of course general types are not per's, but indexed families of per's (plus a relational interpretation of course) so the inclusion of per's into sets must be indexed, and that is the right hand side of the diagram.

The formal definition of the categories of (4) is as follows. The fibre of $\mathbf{Fam}(\mathbf{Set})$ over n has as

Objects maps $f : \mathrm{Obj}(\mathbf{AP}(D))^n \to \mathbf{Set}$.

Morphisms $t : f \to g$ is a family of set theoretic maps

$$(t_{\vec{R}} : f(\vec{R}) \to g(\vec{R}))_{\vec{R}\in\mathrm{Obj}(\mathbf{AP}(D))^n}$$

and reindexing is given by composition. The fibre of $\mathbf{Fam}(\mathrm{Sub}(\mathbf{Set}))$ over an object $f : \mathrm{Obj}(\mathbf{AP}(D))^n \to \mathbf{Set}$ is a preorder with

Objects maps $s : \mathrm{Obj}(\mathbf{AP}(D))^n \to \mathbf{Set}$, such that

$$\forall \vec{R} \in \mathrm{Obj}(\mathbf{AP}(D))^n.\ s(\vec{R}) \subseteq f(\vec{R}).$$

Morphisms There is a morphism $s \to s'$ if

$$\forall \vec{R} \in \mathrm{Obj}(\mathbf{AP}(D))^n.\ s(\vec{R}) \subseteq s'(\vec{R}).$$

Here reindexing with respect to morphisms in $\mathbf{PAP}(D)$ is given by composition, whereas reindexing with respect to morphisms in $\mathbf{Fam}(\mathbf{Set})$ is given by inverse image.

Lemma 4.1. *The fibration s has fibred products, and (r, s) is an indexed first-order logic fibration with simple Ω-products and -coproducts.*

Proof. Clearly s has fibred product inherited from $\mathbf{Set}$. The rest of the lemma states that the fibration r has left and right adjoints to sufficiently many reindexing functors to interpret all the needed quantifications in the logic LAPL. But r is simply an indexed version of the subobject fibration on $\mathbf{Set}$, and since this fibration has left and right adjoints to all reindexing functors, the lemma follows. $\square$

The inclusion functor $I\colon \mathbf{PFam}(\mathbf{AP}(D)) \to \mathbf{Fam}(\mathbf{Set})$ of (4) maps a type (f^p, f^r) to *Classes* $\circ f^p$, and likewise maps a morphism $(t_{\vec{R}})_{\vec{R}}$ to $(\mathit{Classes}(t_{\vec{R}}))_{\vec{R}}$. This corresponds to the intuition described earlier: a type is a pair (f^p, f^r), but when reasoning about a type, we forget the relational interpretation f^r of the type, and reason set theoretically about the equivalence classes of the per.

Lemma 4.2. *I is a faithful and product-preserving map of fibrations.*

As mentioned, I includes the category of per's into a larger category in which the collection of relations between a pair of per's is an object. In the setting of LAPL-structures, this is formulated as a contravariant map of fibrations U:

$$\begin{tikzcd}
\mathbf{PFam}(\mathbf{AP}(D)_\perp)^2 \arrow[rr, "U"] \arrow[dr] & & \mathbf{Fam}(\mathbf{Set}) \arrow[dl] \\
& \mathbf{PAP}(D) &
\end{tikzcd}$$

By contravariant map of fibrations, we mean a map commuting with the reindexing structure, but contravariant in each fibre. The functor U is defined as

$$(f, g) \mapsto 2^{I(f) \times I(g)}.$$

Lemma 4.3. *U is a contravariant map of fibrations.*

The precise formulation of U mapping a pair of types to the collection of all relations on those types is the existence of a family of bijections

$$\chi_n\colon \mathbf{Fam}(\mathbf{Set})(M, U_n(f, g))_n \to \mathrm{Obj}(\mathbf{Fam}(\mathrm{Sub}(\mathbf{Set}))_{M \times I_n(\mathbb{U}_n(f) \times \mathbb{U}_n(g))})$$

indexed over $f, g \in (\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$ and $M \in (\mathbf{Fam}(\mathbf{Set}))_n$. This family is defined by

$$\chi_n(h) = \{(m, (a, b)) | (a, b) \in h(m)\}$$

in other words this is just the usual bijection between set theoretic maps $M_1 \to P(M_2 \times M_3)$ and subsets of $M_1 \times M_2 \times M_3$.

In terms of LAPL-structures we have proved:

Proposition 4.4. *The diagram (4) constitutes a pre-LAPL structure.*

Any type (f^p, f^r) in our model has a relational interpretation given by the map f^r, which we would like to show can be used for reasoning about parametricity. However, f^r is only defined on admissible relations on per's, i.e., not on *any* subset of $\mathit{Classes}(R) \times \mathit{Classes}(S)$. This is no coincidence, as explained in the introduction, and in the logic LAPL [6], axioms are formulated for such a collection of admissible relations to be useful for reasoning about parametricity. We show that the admissible relations used in this paper satisfy these axioms in Lemma 4.5.

First we formulate the collection of admissible relations as a subfunctor V of U by $V(f,g) = \vec{R} \mapsto \{\, \mathit{Classes}(A) \mid A \rightarrowtail_{\mathbf{AP}(D)_\perp} (f^p(\vec{R}) \times g^p(\vec{R}))\,\}$.

Lemma 4.5. *The structure in diagram (4) and V model admissible relations.*

Proof. We must show that the collection of admissible relations used here satisfy the axioms formulated in [6]. Recall that an admissible relation on a pair (R, S) of admissible per's is a regular subobject of the product $R \times S$ in $\mathbf{AP}(D)_\perp$. Since equality is given by the diagonal map, this is admissible, and since regular subobjects are closed under reindexing along maps in $\mathbf{AP}(D)_\perp$, the reindexing axiom is satisfied. By Lemma 2.12, regular subobjects are closed under intersection, which proves that admissible relations are closed under conjunction and universal quantification. Finally, we must show that $(x,y).\, \phi \supset \rho(x,y)$ is admissible if ρ is admissible and ϕ is a proposition, i.e., x, y are not free in ϕ. Since the logic of the pre-LAPL structure (4) is classical set theoretic logic, the proof boils down to the two cases of ϕ being true or false. In the first case we simply get the admissible relation ρ, and in the second we get the total relation $(x,y).\, \top$ which clearly is admissible. □

The final step towards showing that (4) is an LAPL-structure and thus models LAPL, is to show that all types have a relational interpretation. In categorical terms, this is formulated as the existence of a map of fibrations J:

$$\begin{tikzcd} \mathbf{PFam}(\mathbf{AP}(D)_\perp) \arrow[d] & & \mathbf{LinAdmRelations} \arrow[d] \\ \mathbf{PAP}(D) & \arrow[r] & \mathbf{AdmRelCtx} \end{tikzcd}$$

where $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ is a fibration constructed from the pre-LAPL structure (4). Intuitively it is a fibration of relations, and the idea is that J should simply be the map $(f^p, f^r) \mapsto f^r$. We first write out the abstract definition of the fibration of relations in the case of the pre-LAPL structure considered here.

The category $\mathbf{AdmRelCtx}$ has as

Objects triples (n, m, Θ) where $\Theta\colon \mathrm{Obj}(\mathbf{AP}(D))^{n+m} \to \mathbf{Set}$, assigns a set to a vector of admissible per's.

Morphisms triples $(f, g, \rho)\colon (n, m, \Theta) \to (n', m', \Theta')$ where $f\colon n \to n'$ and $g\colon m \to m'$ are morphisms in $\mathbf{PAP}(D)$ and ρ is an indexed family of set theoretic maps

$$\rho = (\rho_{\vec{R},\vec{S}}\colon \Theta(\vec{R}, \vec{S}) \to \Theta'(\vec{f^p}(\vec{R}), \vec{g^p}(\vec{S})))_{\vec{R}\in \mathrm{Obj}(\mathbf{AP}(D))^n, \vec{S}\in \mathrm{Obj}(\mathbf{AP}(D))^m}$$

In this concrete case $\mathbf{LinAdmRelations}$ can be described as follows: Given an object (n, m, Θ) over (n, m), the fibre of $\mathbf{LinAdmRelations}$ over (n, m, Θ) has as

Objects triples (ϕ, f, g) such that f and g are objects of $\mathbf{PFam}(\mathbf{AP}(D)_\perp)$ over n and m respectively and ϕ is an indexed family of maps

$$\phi = (\phi_{\vec{R},\vec{S}} \colon \Theta(\vec{R},\vec{S}) \to \{\, A \mid A \rightarrowtail f^p(\vec{R}) \times g^p(\vec{S}) \,\})_{\vec{R} \in \mathrm{Obj}(\mathbf{AP}(D))^n, \vec{S} \in \mathrm{Obj}(\mathbf{AP}(D))^m}$$

Morphisms A morphism $(\phi, f, g) \to (\psi, f', g')$ is a pair of morphisms

$$(t \colon f \to f', u \colon g \to g')$$

in $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_n$ and $(\mathbf{PFam}(\mathbf{AP}(D)_\perp))_m$, respectively, such that

$$\forall \vec{R} \in \mathrm{Obj}(\mathbf{AP}(D))^n, \vec{S} \in \mathrm{Obj}(\mathbf{AP}(D))^m. \forall M \in \Theta(\vec{R},\vec{S}).$$
$$\forall ([d],[d']) \in \mathit{Classes}(f^p(\vec{R})) \times \mathit{Classes}(g^p(\vec{S})).\, ([d],[d']) \in \phi(M) \Rightarrow (t([d]), u([d'])) \in \psi(M)$$

Notice the two maps of fibrations

$$\begin{tikzcd} \left(\begin{array}{c} \mathbf{PFam}(\mathbf{AP}(D)_\perp) \\ \downarrow \\ \mathbf{PAP}(D) \end{array}\right) & \left(\begin{array}{c} \mathbf{LinAdmRelations} \\ \downarrow \\ \mathbf{AdmRelCtx} \end{array}\right) \arrow[l, shift right, "\partial_0"'] \arrow[l, shift left, "\partial_1"] \end{tikzcd}$$

which on objects of $\mathbf{AdmRelCtx}$ are defined by $\partial_0(n, m, \Theta) = n$ and $\partial_1(n, m, \Theta) = m$ and on objects of $\mathbf{LinAdmRelations}$ map (ϕ, f, g) to f and g respectively. Thinking of $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ as a fibration of relations, these are the maps that map a relation to its domain and codomain respectively.

Finally we can define the required functor J. For the base categories, J is defined on

Objects by $n \mapsto (n, n, (\prod_i \{\, A \mid A \rightarrowtail R_i \times S_i \,\})_{\vec{R},\vec{S} \in \mathbf{AP}(D)^n})$

Morphisms by $f \mapsto (f, f, \prod_i f_i^r)$

and for the total categories, J is defined on

Objects by $(f^p, f^r) \mapsto (f^r, f, f)$

Morphisms by $t \mapsto (t, t)$.

Lemma 4.6. *J is a map of fibrations and $\partial_0 \circ J$ and $\partial_1 \circ J$ are both equal to the identity.*

Lemma 4.7. *J is a map of linear λ_2-fibrations.*

Proof. We must show that J preserves $\multimap$, $\otimes$, $\prod$, I and !. In the fibration $\mathbf{LinAdmRelations} \to \mathbf{AdmRelCtx}$ this structure is defined using syntactic construction on relations. Recall from [6, Remark 2.35] that for $\rho \colon \mathsf{AdmRel}(\sigma, \tau)$, the relation $!\rho$ is the smallest admissible relation containing $(!x, !y)$ whenever $\rho(x, y)$. If $A \rightarrowtail R \times S$ then LA is the smallest regular subobject of $R \times S$ relating the lifts of d and e if d, e are related in A. Since the fibred functor ! on $\mathbf{PFam}(\mathbf{AP}(D)_\perp)$ is defined pointwise by lifting relations, J thus commutes with !. Likewise we can show that J commutes with $\otimes$ using the characterization of $\otimes$ on relations in [6, Remark 2.35] as the smallest admissible relation relating $d \otimes e$ to $d' \otimes e'$ whenever d and d' are related and e and e' are related. The rest of the cases are simple inspections. □

Theorem 4.8. *The diagram in* (4) *constitutes a parametric LAPL-structure.*

Proof. The preceding results show that it is an LAPL-structure; it only remains to show that it is a parametric such. Identity extension holds in the internal language of the LAPL-structure because the relational interpretation of a type is f^r, and this is required to satisfy identity extension. Finally the technical requirements of very strong equality and extensionality hold because the subobject fibration on $\mathbf{Set}$ satisfies very strong equality and extensionality. □

5 Solving recursive type equations

Having shown that (3) extends to a parametric LAPL-structure, the results from [5] apply to our model. In particular, we can solve a large class of recursive domain equations given by a class of fibred functors called strong fibred functors in [5]. The following lemma characterizes strong fibred functors in this concrete model.

Proposition 5.1. *There is a bijective correspondence between strong fibred functors (as defined in [5])* F*:*

$$\begin{tikzcd}
(\mathbf{PFam}(\mathbf{AP}(D)_\perp)^{\mathrm{op}})^n \times \mathbf{PFam}(\mathbf{AP}(D)_\perp)^m \arrow[rr, "F"] \arrow[dr] & & \mathbf{PFam}(\mathbf{AP}(D)_\perp) \arrow[dl] \\
 & \mathbf{PAP}(D) &
\end{tikzcd}$$

and triples (F^p, F^r, F_1)*, where* (F^p, F^r) *is an object of* $(\mathbf{PFam}(\mathbf{AP}(D)_\perp)_{m+n})$ *and* (F^p, F_1) *is a functor* $(\mathbf{AP}(D)_\perp{}^{\mathrm{op}})^n \times \mathbf{AP}(D)_\perp^m \to \mathbf{AP}(D)_\perp$*, and moreover*

- F_1 *has a realizer, i.e., there exists a continuous map* $d\colon [D \to D]^{n+m} \to [D \to D]$ *such that, if the* $\mathbf{AP}(D)_\perp$ *morphisms* $t_1, \dots, t_n, u_1, \dots u_m$ *are realized by* $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m$ *then* $d(\alpha_1 \dots \alpha_n, \beta_1 \dots \beta_m)$ *is a realizer for* $F_1(t_1, \dots, t_n, u_1, \dots, u_m)$*.*
- F_1 *respects relations, i.e., if* $A_i \rightarrowtail R_i \times S_i$ *and* $A'_i \rightarrowtail R'_i \times S'_i$ *and* $(t_i\colon R'_i \to R_i, u_i\colon S'_i \to S_i)$ *preserve relations in the sense that for all* i

  $$\forall([d],[e]) \in A'_i.\, (t_i([d]), u_i([e])) \in A_i$$

  and likewise $B_i \rightarrowtail T_i \times U_i$*,* $B'_i \rightarrowtail T'_i \times U'_i$ *and for all* i *the pair* $(\gamma_i\colon T_i \to T'_i, \delta_i\colon U_i \to U'_i)$ *preserves relations, then also*

  $$(F_1(t_1, \dots, t_n, \gamma_1, \dots, \gamma_m), F_1(u_1, \dots, u_n, \delta_1, \dots, \delta_m))$$

  preserve relations, i.e., $\forall([d],[e]) \in F^r(A_1, \dots, A_n, B_1, \dots, B_m)$

  $$(F_1(t_1, \dots, t_n, \gamma_1, \dots, \gamma_m)([d]), F_1(u_1, \dots, u_n, \delta_1, \dots, \delta_m)([e])) \in F^r(A'_1, \dots, A'_n, B'_1, \dots, B'_m)$$

The main example of a strong fibred functor is the interpretation of a type

$$\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_m \vdash \sigma$$

of pure PILL_Y in which the type variables α_i occur only negatively and the type variables β_i only positively.

Proof. Notice first that $n+m$ is a generic object for the fibration

$$(\mathbf{PFam}(\mathbf{AP}(D)_\perp)^{\mathrm{op}})^n \times \mathbf{PFam}(\mathbf{AP}(D)_\perp)^m \to \mathbf{PAP}(D),$$

and so the object part of a fibred functor F as in the theorem is completely determined by the image on the identity on $n+m$.

If F is a strong fibred functor, then (F^p, F^r) is the image of F applied to the identity on $n+m$, and the existence of the realizer for F_1 follows from the strength of the functor.

For the other direction, suppose we are given (F^p, F^r, F_1) as above. Then the functor F is defined on objects by composition with (F^p, F^r). $\square$

As mentioned, in [5] we prove that all recursive type equations corresponding to strong fibred functors can be solved. For a detailed description of what this means, we refer to *loc. cit.*. Here we mention just the simple case of $n = 0, m = 1$. In this case F is a fibred endofunctor, and since the fibre $\mathbf{PFam}(\mathbf{AP}(D)_\perp)_0$ is isomorphic to $\mathbf{AP}(D)_\perp$ we get the following theorem.

Theorem 5.2. *Suppose (F^p, F^r, F_1) is a strong fibred functor in the case of $n = 1$ and $m = 0$ of Proposition 5.1. Then there exists an admissible per R and an isomorphism $F^p(R) \cong R$ in $\mathbf{AP}(D)_\perp$ which is at the same time an initial algebra and a final coalgebra for the functor $(F^p, F_1)\colon \mathbf{AP}(D)_\perp \to \mathbf{AP}(D)_\perp$.*

6 Example: Natural numbers

As an example of a computation in the model, we compute explicitly the interpretation of the type

$$\prod \alpha.\, (\alpha \multimap \alpha) \to \alpha \multimap \alpha$$

which we know from LAPL is a natural numbers object in $\mathbf{AP}(D)_\perp$ (since this is the fibre of closed types).

Due to shortage of letters in the english alphabet, we will use x, y, f and g in addition to d for elements of D.

To ease notation, given a regular subobject $A \rightarrowtail R \times S$, we shall write $(x, y) \in A$ for $R(x, x)$, $S(y, y)$ and $([x], [y]) \in A$. We will also leave Ψ, Φ implicit, and simply write $f\ x$ for $\Phi(f)(x)$.

We consider the type $\mathrm{Nat} = [\![\prod \alpha.\, (\alpha \multimap \alpha) \to \alpha \multimap \alpha]\!]$. By definition

$$d(\mathrm{Nat}^p)d'$$

iff for all R, S per's and all regular subobjects $A \rightarrowtail R \times S$, $(f, g) \in (A \multimap A)$ and $(x, y) \in A$

$$(d\ f\ x, d'\ g\ y) \in A.$$

The domain of Nat contains the elements $\perp = \lambda f \lambda x.\, \perp$ and $\underline{n} = \lambda f.\, \lambda x.\, f^n(x)$, in particular $\underline{0} = \lambda f \lambda x.\, x$. We also have a map *succ*: $\mathrm{Nat} \to \mathrm{Nat}$ realized by $\lambda n.\, \lambda f.\, \lambda x.\, f(n(f)(x))$, and $succ(\underline{n}) = \underline{n+1}$.

Lemma 6.1. *Suppose $\underline{n} \leq \underline{m}$. Then $n = m$.*

Proof. Consider the two functions $f, g\colon D \to D$ given by $f(d) = \langle d, \mathbf{i} \rangle$, where $\mathbf{i}$ is the code of the identity function, and g being the first projection. Both are continuous and since $g \circ f = id$ f is injective. Define the sequence of elements $x_n = f^n(\perp)$. This sequence is strictly increasing.

Now, if $\underline{n} \leq \underline{m}$ then

$$x_n = \underline{n}\, f \perp \leq \underline{m}\, f \perp = x_m$$

so $n \leq m$. Further,

$$x_{m-n} = \underline{n}\, g\, x_m \leq \underline{m}\, g\, x_m = \perp$$

so $m = n$. □

Lemma 6.2. *The per N given by*

$$\perp N \perp \quad \wedge \quad \forall n \in \mathbb{N}. \underline{n}\, N\, \underline{n}$$

is admissible.

Proof. N has the equivalence classes

$$\{\{\perp\}\} \cup \{\{\underline{n}\} \mid n \in \mathbb{N}\}$$

thus, by the lemma above, there are no interesting chains in N. □

Proposition 6.3. *Suppose $d(\mathrm{Nat}^p)d$. Then*

i) $d = d\, \textit{succ}\, 0$ *and*

ii) either $d = \perp$ *or* $d = \underline{n}$*.*

Proof. Consider the discrete admissible per D:

$$\{\{d\} \mid d \in D\}$$

Then given f, x consider the regular subobject $A \rightarrowtail \mathrm{Nat} \times D$ given by

$$(\perp, \perp) \in A, \qquad \forall n.\, (\underline{n}, f^n(x)) \in A.$$

A is admissible, simply because it contains no interesting increasing chains. Clearly $(\textit{succ}, f) \in A \multimap A$, so

$$(d\, \textit{succ}\, 0, d\, f\, x) \in A,$$

i.e., if $d\, \textit{succ}\, 0 = \perp$, then $d\, f\, x = \perp$ for all f, x and so $d = \perp$, and if $d\, \textit{succ}\, 0 = \underline{n}$ for some n, then $d\, f\, x = f^n(x)$, for all f, x, so $d = \underline{n}$. As we have seen, there are no other possibilities for $d\, \textit{succ}\, 0$. □

Proposition 6.4. *Suppose $d(\mathrm{Nat}^p)d'$, then $d = d'$.*

Proof. By considering the regular subobject $A \rightarrowtail \mathrm{Nat} \times \mathrm{Nat}$ given by

$$(\perp, \perp) \in A, \qquad \forall n.\, (\underline{n}, \underline{n}) \in A$$

we conclude

$$d\, \textit{succ}\, 0 = d'\, \textit{succ}\, 0.$$

By Proposition6.3 part i) we then get $d = d'$. □

In conclusion, by direct calculation we have shown

$$\mathrm{Nat}^p = \{\{\perp\}\} \cup \{\{\underline{n}\} \mid n \in \mathbb{N}\},$$

where the elements $\underline{n}$ are distinct incomparable elements of D.

7 Related PER Models of Recursive Types

As mentioned earlier, the fiber category $\mathbf{PFam}(\mathbf{AP}(D)_{\perp})_0$ is equivalent to $\mathbf{AP}(D)_{\perp}$. Hence the results on solutions to recursive domain equations of Section 5 imply that we can solve a wide class of recursive domain equations on $\mathbf{AP}(D)_{\perp}$. In other words, our abstract results show that admissible per's provide a model of recursive types. Previous per models of recursive types, however, have involved extra conditions on the per's beyond admissibility.

In [1] a per model of polymorphism and recursive types is constructed. It employs per's, which are admissible, meet closed, uniform and convex. An O-category of these so-called *good* per's is constructed and type expressions can now be modeled as effective symmetric functors on this category. In [2] it is shown how complete uniform per's (cuper's) over a universal domain allows one to solve domain equations on the per level. In both cases the chosen notion of per's facilitate an ordering of the equivalence classes and thus allows one to solve recursive domain equations as in classical domain theory.

In [1] the collection of domain equations that can be solved are given by the notion of effective symmetric functors. Comparing these with the strong fibred functors of our setting we see that both notions require a realizer, but our functors are also required to have a relational interpretation given by the component F^r as in Lemma 5.1. It appears that our notion of recursive type equations are more restrictive, but on the other hand our notion of admissible per's is simpler. We find this trade-off acceptable, as all type expressions formed using the type constructors of Polymorphic FPC give rise to a strong fibred functor [15]. The real difference, however, is that our model is parametric.

Acknowledgments

We gratefully acknowledge discussions with Milly Maietti, Gordon Plotkin, John Reynolds, Pino Rosolini and Alex Simpson.

References

[1] M. Abadi and G.D. Plotkin. A per model of polymorphism and recursive types. In *5th Annual IEEE Symposium on Logic in Computer Science*, pages 355–365. IEEE Computer Society Press, 1990. 7

[2] Roberto M. Amadio and Pierre-Louis Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1998. 2, 7

[3] A. Barber. *Linear Type Theories, Semantics and Action Calculi*. PhD thesis, Edinburgh University, 1997. 3, 3

[4] G. M. Bierman, A. M. Pitts, and C. V. Russo. Operational properties of Lily, a polymorphic linear lambda calculus with recursion. In *Fourth International Workshop on Higher Order Operational Techniques in Semantics, Montréal*, volume 41 of *Electronic Notes in Theoretical Computer Science*. Elsevier, September 2000. 1

[5] L. Birkedal, R. E. Møgelberg, and R. L. Petersen. Category theoretic models of linear Abadi & Plotkin logic. 2006. Submitted. (document), 1, 4, 5, 5.1, 5

[6] L. Birkedal, R. E. Møgelberg, and R. L. Petersen. Linear Abadi & Plotkin logic. *Logical Methods in Computer Science*, 2(5:1):1–33, 2006. 1, 3, 4, 4, 4, 4

[7] L. Birkedal, R.E. Møgelberg, and R.L. Petersen. Parametric domain-theoretic models of polymorphic intuitionistic / linear lambda calculus. In M. Escardó, A. Jung, and M. Mislove, editors, *Proceedings of Mathematical Foundations of Programming Semantics 2005*, volume 155, pages 191–217, 2005. (document), 1

[8] M. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps*. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996. 1

[9] J.-Y. Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur*. Thèse d'Etat, Université Paris VII, 1972. 1

[10] M. Hasegawa. Logical predicates for intuitionistic linear type theories. In *Proc. 4th International Conference on Typed Lambda Calculi and Applications (TLCA'99)*, volume 1581 of *Lecture Notes in Computer Science*, pages 198–213. Springer, 1999. 2.1

[11] H. Huwig and A. Poigné. A note on inconsistencies caused by fixpoints in a cartesian closed category. *Theoretical Computer Science*, 73:101–112, 1990. 1

[12] B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69:73–106, 1994. 2.2

[13] B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishers B.V., 1999. 3, 3, 3.1

[14] R. E. Møgelberg. *Category theoretic and domain theoretic models of parametric polymorphism*. PhD thesis, IT University of Copenhagen, 2005. 3.1

[15] R. E. Møgelberg. Interpreting polymorphic FPC into domain theoretic models of parametric polymorphism. In *ICALP: Annual International Colloquium on Automata, Languages and Programming*, 2006. To appear. 1, 7

[16] R. E. Møgelberg, L. Birkedal, and R. L. Petersen. Categorical models of PILL. Technical Report TR-2005-58, IT University of Copenhagen, February 2005. 2.1, 3, 3

[17] B.C. Pierce. *Types and Programming Languages*. MIT Press, 2002. 1

[18] A. M. Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in computer Science*, 10:321–359, 2000. 1

[19] G.D. Plotkin. Lectures on predomains and partial functions. Notes for a course given at the Center for the Study of Language and Information, Stanford, 1985. 1

[20] G.D. Plotkin. Second order type theory and recursion. Notes for a talk at the Scott Fest, February 1993. 1

[21] Gordon Plotkin and Martín Abadi. A logic for parametric polymorphism. In *Typed lambda calculi and applications (Utrecht, 1993)*, volume 664 of *Lecture Notes in Comput. Sci.*, pages 361–375. Springer, Berlin, 1993. 1

[22] J.C. Reynolds. Towards a theory of type structure. In *Colloquium sur La Programmation*, volume 19 of *Lecture Notes in Computer Science*, pages 408–423. Springer-Verlag, 1974. 1

[23] J.C. Reynolds. Types, abstraction, and parametric polymorphism. *Information Processing*, 83:513–523, 1983. 1

[24] J.C. Reynolds. Private communication, June 2000. 1

[25] E.P. Robinson and G. Rosolini. Reflexive graphs and parametric polymorphism. In S. Abramsky, editor, *Proc. 9th Symposium in Logic in Computer Science*, pages 364–371, Paris, 1994. I.E.E.E. Computer Society. 1, 3.1

[26] G. Rosolini and A. Simpson. Using synthetic domain theory to prove operational properties of a polymorphic programming language based on strictness. Manuscript, 2004. 1