

DOMAIN THEORETIC MODELS OF POLYMORPHISM

**Thierry Coquand, Carl A. Gunter
and Glynn Winskel**

**MS-CIS-88-38
LINC LAB 115**

**Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104**

June 1988

To appear in a forth coming issue of the "Information and Computation "

Acknowledgements: This research was supported in part by DARPA grant NOOO14-85-K-0018, NSF grants MCS-8219196-CER, IRI84-10413-AO2 and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.

DOMAIN THEORETIC MODELS OF POLYMORPHISM*

Thierry Coquand[†]
INRIA Rocquencourt

Carl A. Gunter[‡]
University of Pennsylvania

Glynn Winskel[§]
Aarhus University

June 6, 1988

Abstract

We give an illustration of a construction useful in producing and describing models of Girard and Reynolds' polymorphic λ -calculus. The key unifying ideas are that of a Grothendieck fibration and the category of continuous sections associated with it, constructions used in indexed category theory; the universal types of the calculus are interpreted as the category of continuous sections of the fibration. As a major example a new model for the polymorphic λ -calculus is presented. In it a type is interpreted as a Scott domain. In fact, understanding universal types of the polymorphic λ -calculus as categories of continuous sections appears to be useful generally. For example, the technique also applies to the finitary projection model of Bruce and Longo, and a recent model of Girard. (Indeed the work here was inspired by Girard's and arose through trying to extend the construction of his model to Scott domains.) It is hoped that by pin-pointing a key construction this paper will help towards a deeper understanding of models for the polymorphic λ -calculus and the relations between them.

1 Introduction.

Jean-Yves Girard presented his discovery of the polymorphic λ -calculus in the paper [7]. His motivations came from proof-theory and his use of the calculus to represent proofs in second-order arithmetic. Later, in [21], John Reynolds rediscovered the calculus independently though his motivation was different, being to provide a formal basis to certain polymorphic type disciplines in programming languages. In designing the calculus, Girard and Reynolds each extended the typed λ -calculus to allow a form of parametric polymorphism. Types include universal types which are types of polymorphic terms, thought of as describing those functions which are defined in a uniform manner at all types. Terms can be applied to types and in this sense can be parameterised by types.

In more detail, type variables α are introduced into the typed λ -calculus so, for instance, $\lambda x : \alpha. x$ should be thought of as the identity function on the type denoted by α . The polymorphic identity function, the term which denotes the identity function on any type, is denoted by the term $\Lambda \alpha. \lambda x : \alpha. x$. It has

*To appear in *Information and Computation*.

[†]B.P. 105, 78150 Le Chesnay Cedex, France

[‡]Department of Computer and Information Sciences, Philadelphia, PA 19104 U.S.A.

[§]Computer Science Department, Ny Munkegade, DK-8000 Aarhus C, Denmark.

a universal type denoted by $\Pi\alpha.\alpha \rightarrow \alpha$. Given a type σ_1 , a term $\Lambda\alpha.t$ of universal type $\Pi\alpha.\sigma_2$ can be instantiated to a term $[\sigma_1/\alpha]t$ which then has type $[\sigma_1/\alpha]\sigma_2$, and so, for instance, the polymorphic identity above instantiates at type σ to the identity $\lambda x : \sigma.x$ of type $\sigma \rightarrow \sigma$.

While the pioneering work of Girard contains most of the results on the syntax of the calculus, an understanding of its models and semantics has developed more slowly and is still incomplete. There is a trivial model got by interpreting types as either the empty or one-point set. While from a proof-theoretic view there may be some use in this when the one-point set represents true and the empty set false (*e.g.* to prove consistency as in [25]), it is clearly inadequate as a model of polymorphism. In essence, the difficulty of providing nontrivial models arises from the impredicative nature of the calculus; in the abstraction of a universal type $\Pi\alpha.\sigma$ the type-variable α is understood to range over all types including the universal type itself. This makes it impossible to interpret types as nontrivial sets in a classical set theory (see [20]) although, lately, Pitts has shown how polymorphism can be interpreted in a constructive set theory [18]. Until recently the only nontrivial models known were either term models or realisability models [7] or, following ideas of McCracken [17] and Scott, models based on a universal domain in which types are coded-up as particular kinds of retracts. The latter are models for stronger calculi with a type of types and so are not tailored directly to the requirements of polymorphic λ -calculus and do not in themselves suggest a general definition of model for the calculus. In his paper [8], Girard produced an interesting new model in which types of the polymorphic λ -calculus are represented as certain kinds of objects called qualitative domains, work which was extended in [4]. The category of domains used in [8] and [4] is not the usual one taken in denotational semantics—in particular the morphisms are functions which are stable in the sense of Berry and not just Scott continuous. The work left open the question of whether or not a model similar to Girard’s could be found in the more traditional category of Scott domains and continuous functions.

One achievement of this paper is to present such a model for the polymorphic λ -calculus. It can be viewed as doing with Scott domains and continuous functions what Girard did with qualitative domains and stable functions. Types will be interpreted as Scott domains and types with free type variables, called “variable types” by Girard, as continuous functors on a category of Scott domains. Although Girard’s work provided inspiration, the construction of domains to denote universal types is different.

We have taken trouble to expose the abstract construction of which our model is an instance. A key unifying idea is that of a Grothendieck fibration and the category of its continuous sections. A universal type is interpreted as a category (in this case a domain) of continuous sections of a fibration. Looked at in this way, Girard’s construction, the retract models of McCracken and Scott, and the construction here are all based on instances of a common idea, that universal types are interpreted as continuous sections of a Grothendieck fibration.

We briefly outline the paper. The following section, section 2, introduces the basic ideas of domain theory and category theory on which we shall rely. Section 3 contains a treatment of Grothendieck fibrations and continuous sections, instances of which are given for domains; taking the base category to be a domain we obtain constructions to represent the dependent sum and product types as used in, *e.g.*, Martin-Löf type theory while taking a suitable category of domains as the base category we get a construction we shall use later as the denotation of universal types. For concreteness, we show how the

construction can be carried out in the framework of information systems—an elementary representation of domains. Section 4 contains proofs of several of the technical lemmas needed for the demonstration that our construction yields a model of the polymorphic λ -calculus. Section 5 gives the syntax of the polymorphic λ -calculus with its equational rules and Section 6 its denotational semantics accompanied by proofs of the soundness of the rules. In section 7 where we show how the traditional domain models of polymorphism of McCracken and Scott using retracts can be cast in this light (very similar ideas appear in the thesis work of Taylor, [29]). Finally, in the conclusion, we present our views on the state of the art of models for polymorphism.

As we have already stated the work of Girard has been a guiding influence on this work. We have received encouragement and advice from a number of people whom we thank; we are grateful to Martin Hyland for pointing-out that a construction we produced could be based on a Grothendieck fibration, to Eugenio Moggi for the remark that this construction applied to Girard’s model as well, and to Pino Rosolini for valuable discussions. The significance of fibrations in modelling polymorphism has been anticipated in the thesis work of Paul Taylor (see [29]) who gave a category-theoretic analysis of the concept of a type of types using indexed category theory (but exclusively, it seems, considering domains indexed by partial orders and not as here by categories of embeddings).

2 Categories and domains.

In this section we review basic concepts from category and domain theory. Its purpose is largely to establish notation and terminology. We assume the reader has some familiarity with these topics. A knowledge of the results in [28] would be a good starting point; most of the proofs for results stated in this section can be found there.

Let $\langle I, \leq \rangle$ be a partial order. We say that I is *directed* if it is nonempty and, for any i and j in I , there is a $k \in I$ such that $i \leq k$ and $j \leq k$. A partial order $\langle D, \leq \rangle$ having a least element \perp is said to be *complete* (and we say that D is a complete partial order, abbreviated to *cpo*) if every directed subset $M \subseteq D$ has a least upper bound $\bigvee D$. A point x of a cpo D is said to be *finite* if, for every directed collection $M \subseteq D$ such that $x \leq \bigvee M$, there is a $y \in M$ such that $x \leq y$. Let \mathbf{B}_D denote the collection of finite elements of D . The cpo D is *algebraic* if, for every $x \in D$, the set $M = \{x_0 \in \mathbf{B}_D \mid x_0 \leq x\}$ is directed and $x = \bigvee M$. A cpo D is *bounded complete* if every bounded subset of D has a least upper bound. We call bounded algebraic cpo’s *Scott domains* or just *domains*. In a domain, least upper bounds of finite sets of finite elements are finite, when they exist.

A function $f : D \rightarrow E$ between cpo’s D and E is *monotonic* if it is order preserving, *i.e.* if $x \leq y$ then $f(x) \leq f(y)$. A monotonic function $f : D \rightarrow E$ between cpo’s D and E is *continuous* if $f(\bigvee M) = \bigvee f(M)$ for any directed $M \subseteq D$. Domains with continuous functions form a category \mathbf{D} which is very important for denotational semantics. It is cartesian-closed. Let D and E be two domains. Their product is the domain $D \times E$ consisting of pairs of elements ordered coordinatewise, with the obvious projections. Their function space $D \rightarrow E$ consists of the continuous functions from D to E

ordered pointwise, sometimes called the extensional order, *i.e.*

$$f \leq g \text{ iff } \forall d \in D. f(d) \leq g(d).$$

A pair of continuous functions (f, g) , with $f : D \rightarrow E$ and $g : E \rightarrow D$ between cpo's D, E , is said to be an *embedding-projection pair* if $g \circ f(d) = d$, for all $d \in D$, and $f \circ g(e) \leq e$, for all $e \in E$; then f is called the *embedding* and g the *projection*. We use equally the notations $f \circ g$ or fg for the composition of functions, and use the following notation to pick out the embedding and projection parts of an embedding-projection pair $h = (f, g)$: let $h^L = f$ and $h^R = g$. We remark that as embedding-projection pairs are an example of an adjunction, in this case between very simple partial order categories, it follows that an embedding determines its accompanying projection uniquely and *vice versa*. The category of domains with embedding-projection pairs as morphisms will be of central importance to us. We call the category \mathbf{D}^{EP} , and write $h \in \mathbf{D}^{\text{EP}}(D, E)$ to mean h is an embedding-projection pair, with embedding part a function $h^L : D \rightarrow E$. We take the composition of two embedding-projection pairs $h = (h^L, h^R) \in \mathbf{D}^{\text{EP}}(D, E)$ and $k = (k^L, k^R) \in \mathbf{D}^{\text{EP}}(E, F)$ to be $k \circ h = (k^L \circ h^L, h^R \circ k^R) \in \mathbf{D}^{\text{EP}}(D, F)$. The identity of a domain D in this category is the pair $(\text{id}_D, \text{id}_D)$.

A partial order $\langle I, \leq \rangle$ forms a category in which the objects are the elements of I and the set of morphisms from point x to point y , written $D(x, y)$, is a one point set when $x \leq y$ and is empty otherwise. A directed family in \mathbf{D}^{EP} consists of a functor from a directed set $\langle I, \leq \rangle$ to \mathbf{D}^{EP} ; as such it provides an indexing of a family of objects $D_i \in \mathbf{D}^{\text{EP}}$, for $i \in I$, and morphisms $f_{ij} \in \mathbf{D}^{\text{EP}}(D_i, D_j)$, for $i \leq j$, so that $f_{ii} = \text{id}_{D_i}$ and $f_{ik} = f_{jk} \circ f_{ij}$ whenever $i \leq j \leq k$. A cone for such a directed family is a family of morphisms $\langle \rho_i \in \mathbf{D}^{\text{EP}}(D_i, D) \rangle_{i \in I}$, for a domain D , such that $\rho_i = \rho_j \circ f_{ij}$ for all $i, j \in I$. Note that because embeddings are monic the morphisms f_{ij} of the directed family are uniquely determined by the cone. And in future we shall most often speak of a cone for a directed family without troubling to mention the directed family of which it is a cone; this will always be understood to be that uniquely determined directed family with morphisms $f_{ij} = \rho_j^R \circ \rho_i^L$, for $i, j \in I$. A *directed colimit* is a cone $\langle \rho_i \in \mathbf{D}^{\text{EP}}(D_i, D) \rangle_{i \in I}$ for a directed family, with the universal property that for any other cone, $\langle \rho'_i \in \mathbf{D}^{\text{EP}}(D_i, D') \rangle_{i \in I}$, there is a unique mediating morphism $h \in \mathbf{D}^{\text{EP}}(D, D')$ such that $\rho'_i = \rho_i \circ h$ for all $i \in I$. That is, an initial object in the category of cones. In general, we say that a category \mathbf{C} is *directed complete* if it has colimits for all directed families. So, in particular, a cpo is directed complete when regarded as a category.

The category \mathbf{D}^{EP} is another example of a directed complete category, and we shall often be concerned with calculations involving its directed colimits. It will be useful to relate embedding-projection pairs into a common domain D via certain morphisms in $\mathbf{D}^{\text{EP}}(D, D)$ which correspond to the images of the embeddings in D .

Lemma 1 *Let X, Y, D be domains. Let $f \in \mathbf{D}^{\text{EP}}(X, D)$ and $g \in \mathbf{D}^{\text{EP}}(Y, D)$. Then*

$$(g^R \circ f^L, f^R \circ g^L) \in \mathbf{D}^{\text{EP}}(X, Y) \text{ iff } f^L \circ f^R \leq g^L \circ g^R. \blacksquare$$

Theorem 2 *The category \mathbf{D}^{EP} is directed complete. A cone $\langle \rho_i \in \mathbf{D}^{\text{EP}}(D_i, D) \rangle_{i \in I}$ is a directed colimit iff $\{\rho_i^L \circ \rho_i^R \mid i \in I\}$ is directed in $D \rightarrow D$ and*

$$\text{id}_D = \bigvee \{\rho_i^L \circ \rho_i^R \mid i \in I\}. \blacksquare$$

Theorem 3 *Let D be a domain. Then*

$$\{f^L \circ f^R \mid f \in \mathbf{D}^{\text{EP}}(X, D) \text{ for some finite } X\}$$

is a directed subset of finite elements in $D \rightarrow D$ and

$$\text{id}_D = \bigvee \{f^L \circ f^R \mid f \in \mathbf{D}^{\text{EP}}(X, D) \text{ for some finite } X\}. \blacksquare$$

By virtue of Theorem 2 we see Theorem 3 implies that a domain is the colimit of the finite domains which embed into it. From the fact that the set in the theorem is directed we deduce the following:

Lemma 4 *Let $f_0 \in \mathbf{D}^{\text{EP}}(X_0, D)$ and $f_1 \in \mathbf{D}^{\text{EP}}(X_1, D)$ where X_0, X_1 are finite domains. Then there is a finite domain X and $g \in \mathbf{D}^{\text{EP}}(X, D)$ so that $g_0 = \langle g^R \circ f_0^L, f_0^R \circ g^L \rangle \in \mathbf{D}^{\text{EP}}(X_0, X)$ and $g_1 = \langle g^R \circ f_1^L, f_1^R \circ g^L \rangle \in \mathbf{D}^{\text{EP}}(X_1, X)$ with $f_0 = gg_0$ and $f_1 = gg_1$. \blacksquare*

From the fact that the elements in the set in Theorem 3 are finite we deduce:

Lemma 5 *Suppose $\langle \rho_i \in \mathbf{D}^{\text{EP}}(D_i, D) \rangle_{i \in I}$ is a directed colimit in \mathbf{D}^{EP} . If X is a finite domain and $f \in \mathbf{D}^{\text{EP}}(X, D)$ then there is some $i \in I$ and $h \in \mathbf{D}^{\text{EP}}(X, D_i)$ such that $f = \rho_i \circ h$. \blacksquare*

Given categories \mathbf{C} and \mathbf{C}' , we define the *product category* $\mathbf{C} \times \mathbf{C}'$ to be the category which has as objects pairs (C, C') where C and C' are objects of \mathbf{C} and \mathbf{C}' respectively. The arrows are pairs $(f, g) : (X, X') \rightarrow (Y, Y')$ where $f \in \mathbf{C}(X, Y)$ and $g \in \mathbf{C}'(X', Y')$ with the obvious composition and identity. There are also projections

$$\text{Fst}_{\mathbf{C}, \mathbf{C}'} : \mathbf{C} \times \mathbf{C}' \rightarrow \mathbf{C}$$

$$\text{Snd}_{\mathbf{C}, \mathbf{C}'} : \mathbf{C} \times \mathbf{C}' \rightarrow \mathbf{C}'.$$

When understood from context, the subscripts will usually be dropped. If $F_1 : \mathbf{C} \rightarrow \mathbf{C}_1$ and $F_2 : \mathbf{C} \rightarrow \mathbf{C}_2$ are functors, then there is a unique functor $\langle F_1, F_2 \rangle : \mathbf{C} \rightarrow \mathbf{C}_1 \times \mathbf{C}_2$ such that $\text{Fst} \circ \langle F_1, F_2 \rangle = F_1$ and $\text{Snd} \circ \langle F_1, F_2 \rangle = F_2$. In particular, the diagonal functor $\Delta : \mathbf{C} \rightarrow \mathbf{C} \times \mathbf{C}$ is $\langle \text{id}_{\mathbf{C}}, \text{id}_{\mathbf{C}} \rangle$. If $F : \mathbf{C}_1 \rightarrow \mathbf{C}_2$ and $F' : \mathbf{C}'_1 \rightarrow \mathbf{C}'_2$ then we define

$$F \times G = \langle F \circ \text{Fst}, G \circ \text{Snd} \rangle : \mathbf{C}_1 \times \mathbf{C}_2 \rightarrow \mathbf{C}'_1 \times \mathbf{C}'_2.$$

We write 1 for the terminal category which has one object and one arrow and $1_{\mathbf{C}}$ for the unique functor from a category \mathbf{C} to 1 . Given a category \mathbf{C} and a number $n \geq 0$, we define the n 'th power \mathbf{C}^n of \mathbf{C} by taking $\mathbf{C}^0 = 1$ and $\mathbf{C}^{n+1} = \mathbf{C}^n \times \mathbf{C}$. More generally, we define the *multiary product* of a list of categories by setting $\times() = 1$ and $\times(\mathbf{C}_1, \dots, \mathbf{C}_{n+1}) = (\times(\mathbf{C}_1, \dots, \mathbf{C}_n)) \times \mathbf{C}_{n+1}$.

A functor $F : \mathbf{C} \rightarrow \mathbf{C}'$ between directed complete categories \mathbf{C} and \mathbf{C}' is *continuous* just in case it preserves directed colimits. A continuous function is thus an example of a continuous functor on categories which are partial orders. It is easy to check that a functor $F : \mathbf{C}_1 \times \mathbf{C}_2 \rightarrow \mathbf{C}$ is continuous iff it is continuous in each of its arguments individually. As our categories \mathbf{C} will often have the form $(\mathbf{D}^{\text{EP}})^m$ the problem of verifying continuity we often reduce to the problem of whether or not functors $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ are continuous. To verify the continuity of such a functor it is very useful to employ the following:

Lemma 6 A functor $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ is continuous iff whenever X is a domain and there is a family of domains X_i and functions $f_i \in \mathbf{D}^{\text{EP}}(X_i, X)$, such that $\{f_i^L \circ f_i^R \mid i \in I\}$ is directed and $\bigvee_i f_i^L \circ f_i^R = \text{id}_X$, then $\bigvee_i F^L(f_i) \circ F^R(f_i) = \text{id}_{F(X)}$. ■

The product operator \times on categories cuts down to a continuous functor

$$\times : \mathbf{D}^{\text{EP}} \times \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}.$$

When D and E are domains, we write id_D , $\text{fst}_{D,E}$ and $\text{snd}_{D,E}$ rather than Id_D , $\text{Fst}_{D,E}$ and $\text{Snd}_{D,E}$. The function space operator \rightarrow is also a functor on \mathbf{D}^{EP} . Suppose $f \in \mathbf{D}^{\text{EP}}(X, X')$ and $g \in \mathbf{D}^{\text{EP}}(Y, Y')$. Then we define $f \rightarrow g \in \mathbf{D}^{\text{EP}}(X \rightarrow Y, X' \rightarrow Y')$ by setting

$$(f \rightarrow g)^L(h) = g^L \circ h \circ f^R$$

for $h \in \mathbf{D}(X, Y)$ and

$$(f \rightarrow g)^R(h') = g^R \circ h' \circ f^L$$

for $h' \in \mathbf{D}(X', Y')$.

When functors on \mathbf{D}^{EP} take several arguments we can make their manipulation a little tidier by introducing the following notation. Given a functor $F : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$, we define a functor $F^L : \mathbf{C} \rightarrow \mathbf{D}$ as follows. The action of F^L on objects of \mathbf{C} is the same as F . Given a function $f \in \mathbf{C}(X, Y)$, we define $F^L(f) = (F(f))^L \in \mathbf{D}(F(X), F(Y))$. We also define a functor $F^R : \mathbf{C}^{\text{op}} \rightarrow \mathbf{D}$ by taking the action of F^R on objects to be that of F and defining $F^R(f) = (F(f))^R \in \mathbf{D}(F(Y), F(X))$. We may also write $(Ff)^R$ or even $F(f)^R$ when the meaning is clear from context.

In our semantic treatment of type expressions we will have to cope with the presence of free type-variables and a type expression will denote a functor whose arguments provide an environment associating values with these variables. It is convenient to define generalisations of the product and function space functors on \mathbf{D}^{EP} to cope with these extra parameters. Given functors $F : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$ and $G : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$ we define

$$F \# G = \times \circ (F \times G) \circ \Delta : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$$

$$F \Rightarrow G = \rightarrow \circ (F \times G) \circ \Delta : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$$

We also define a multiary version of the $\#$ operation by taking $\#()$ to be the functor $1_{\mathbf{C}}$ into the trivial domain and setting $\#(F_1, \dots, F_{n+1}) = \#(F_1, \dots, F_n) \# F_{n+1}$. Given functors F_1, \dots, F_n and numbers $n \geq i \geq 1$, we define i 'th projection

$$\mathbf{p}_X^{i,n} : \times(F_1(X), \dots, F_n(X)) \rightarrow F_i(X)$$

by taking

$$\mathbf{p}_X^{i,n} = \begin{cases} \text{fst}_{\times(F_1(X), \dots, F_{n-1}(X)), F_n(X)} \circ \mathbf{p}_X^{i,n-1} & \text{if } i < n \\ \text{snd}_{\times(F_1(X), \dots, F_{n-1}(X)), F_n(X)} & \text{otherwise.} \end{cases}$$

To keep the number of parentheses to a minimum in the calculations we make, it is helpful to introduce some binding conventions. We will assume that association is to the left, so an expression such as $fx y$ or $f(x)(y)$ will be parsed as $(f(x))(y)$. This convention also holds for the application of a section to

an object; so $f(t)_X$ parses as $(f(t))_X$. However, we read an expression such as $t_{G(X)}$ as $t_{(G(X))}$ so that $ft_{G(X)}$ parses as $(f(t))_{(G(X))}$. We assume that application binds more tightly than composition; so $F^R(f) \circ F^R(g)$ parses as $(F^R(f)) \circ (F^R(g))$ and $f \circ t_X$ parses as $f \circ (t_X)$. For functors, we assume that $\#$ binds more tightly than \Rightarrow , so that $F_1 \# F_2 \Rightarrow F$ parses as $(F_1 \# F_2) \Rightarrow F$. We assume that Π^m (introduced in section 3) binds more tightly than either $\#$ or \Rightarrow . Application will bind more tightly than \times or \rightarrow , so that $F(X) \times G(X)$ parses as $(F(X)) \times (G(X))$.

3 Interpreting types.

In our approach, closed types (those with no free type variables) will denote domains. Types with free variables will denote functors on domains which yield a domain once they are given an instantiation of their free variables. Thought of in this way the denotation of a type $\Pi\alpha.\sigma$ should be a functor taking one less argument than that for σ in a way which respects the rules of the polymorphic λ -calculus. In this section we work towards the definition of an operation on functors to achieve this. The operation, again called Π , shares many properties with universal quantification, and indeed can be viewed abstractly in a similar way, as right adjoint to the operation of “padding out” a functor with an extra argument. Our treatment conforms to the category-theoretic definition of model for the polymorphic λ -calculus proposed by Seely [24], though for the most part we shall express our ideas concretely, through giving particular constructions on domains. Our more concrete approach will, however, be enough here (in the same way that it is not necessary to know what a cartesian-closed category is in order to understand what it means to be a model of simple typed lambda-calculus). A slight exception to this approach arises in the construction of Π which we show is a special case of a general one, traditional in category theory, that of sections of the Grothendieck fibration of a functor. Other familiar constructions on types like dependent sum and product arise as special cases too.

3.1 Fibrations and sections.

Let $F : \mathbf{C} \rightarrow \mathbf{Cat}$ be a continuous functor from a category \mathbf{C} to the category of all categories. Define the *Grothendieck fibration* of F to be the category ΣF consisting of

- *objects* which are pairs (X, t_X) where $X \in \mathbf{C}$ and $t_X \in F(X)$, and
- *morphisms* $(X, t_X) \rightarrow (Y, t_Y)$ which are pairs (f, α) where $f : X \rightarrow Y$ in \mathbf{C} and $\alpha : F(f)(t_X) \rightarrow t_Y$ in $F(Y)$

with the *composition* of two morphisms $(f, \alpha) : (X, t_X) \rightarrow (Y, t_Y)$ and $(g, \beta) : (Y, t_Y) \rightarrow (Z, t_Z)$ given by

$$(g, \beta) \circ (f, \alpha) = (g \circ f, \beta \circ F(g)(\alpha)).$$

Then ΣF is a category with the identity morphism on (X, t_X) being $(\text{id}_X, \text{id}_{t_X})$.

The *projection* $p : \Sigma F \rightarrow \mathbf{C}$ is defined to be the functor which takes $(f, \alpha) : (X, t_X) \rightarrow (Y, t_Y)$ to $f : X \rightarrow Y$.

We remark that our definition of Grothendieck fibration is not quite standard as it is traditional to work with opposite categories and, consequently, have the functor F take arguments in a category \mathbf{C}^{op} (so that cofibration would perhaps be a better name); for our purposes this would be inconvenient.

The construction ΠF has continuous sections as objects. A *section* of ΣF is a functor $s : \mathbf{C} \rightarrow \Sigma F$ such that $p \circ s = \text{id}_{\mathbf{C}}$, and, of course, a continuous section is such a functor which is continuous. Taking sections as objects we form a category by taking morphisms to be cartesian natural transformations, *i.e.* those natural transformations which project under p to identity morphisms in \mathbf{C} . A typical morphism between sections is a natural transformation ν from a section s to section s' consisting of a family $\langle \nu_X \rangle_{X \in \mathbf{C}}$ of morphisms $\nu_X : s(X) \rightarrow s'(X)$ in ΣF where $p(\nu_X) = \text{id}_X$ for all $X \in \mathbf{C}$. Of course, each component ν_X of such a natural transformation must have the form $\nu_X = (\text{id}_X, \alpha_X)$ with $\alpha_X : t_X \rightarrow t'_X$ where $s(X) = (X, t_X)$ and $s'(X) = (X, t'_X)$. Being a natural transformation ensures that for all $f : X \rightarrow Y$ we have $\nu_Y \circ s(f) = s'(f) \circ \nu_X$. The category ΠF is defined to be the full subcategory of continuous sections.

3.2 Families indexed by a domain.

We shall be concerned with fibrations and sections solely for the case in which the functor F takes values which are domains. Then for special forms of base category \mathbf{C} the structure ΠF , in general a category, will be isomorphic to a domain. A simple example arises when \mathbf{C} is a domain itself and the functor F goes from the domain to the category of domains with embeddings; in this case not only is ΠF a domain but so is ΣF . We shall call these constructions *dependent product* and *dependent sum*, following the terminology in Martin-Löf type theory [14], [15]. (The constructions seem to be well-known and appear in the exercises of [19].) A more abstract presentation would have been to use the ideas of [24] in order to give a categorical characterisation of the dependent product and sums, and to show that the constructions we give verify these properties (see also [5]). See section 7 for an application of dependent products.

Let C be a domain regarded as a category so there is a unique morphism from x to y precisely when $x \leq y$; thinking of the graph of the order relation as being the set of morphisms, we shall write (x, y) for the unique morphism from x to y . Let $F : C \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor to the category of domains with embedding-projection pairs. The functor F provides a domain $F(x)$ for each element x of C and embeddings $F(x, y)^L : F(x) \rightarrow F(y)$ for $x \leq y$ in C . These satisfy the functor laws so $F(x, x)^L = \text{id}_{F(x)}$ and if $x \leq y \leq z$ then $F(x, z)^L = F(y, z)^L \circ F(x, y)^L$. In this case the category ΣF has objects (x, t_x) where $x \in C$ and $t_x \in F(x)$. A morphism $(x, t_x) \rightarrow (y, t_y)$ arises when and only when $x \leq y$ in C and $F(x, y)^L(t_x) \leq t_y$ in $F(y)$. It follows that the category ΣF is isomorphic to a partial order defined on objects of ΣF by

$$(x, t_x) \leq (y, t_y) \text{ iff } x \leq y \text{ and } F(x, y)^L(t_x) \leq t_y,$$

It is easy to check this relation is a partial order, and, perhaps not surprisingly, ΣF is a domain too.

Proposition 7 *Let C be a domain. Let $F : C \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor. Then ΣF is a domain. In this case the projection functor is a continuous function $p : \Sigma F \rightarrow C$ between domains.*

Proof: ΣF has a least element $(\perp, \perp_{F(\perp)})$. Suppose $V = \{(x_i, t_i) \mid i \in I\}$ is a directed subset of ΣF . Then $\{x_i \mid i \in I\}$ is a directed subset of C and so has a least upper bound $x = \bigvee_{i \in I} x_i$ in C . It is easy to see the set $\{F(x_i, x)^L(t_i) \mid i \in I\}$ is directed. Taking $t = \bigvee_{i \in I} F(x_i, x)^L(t_i)$ we show that (x, t) is the least upper bound of V in ΣF . Clearly it is an upper bound and supposing $(x_i, t_i) \leq (x', t')$, for all $i \in I$, we see $x \leq x'$ and $F(x_i, x')^L(t_i) \leq t'$ for all $i \in I$ whence

$$\begin{aligned} F(x, x')^L(t) &= F(x, x')^L\left(\bigvee_{i \in I} F(x_i, x)^L(t_i)\right) \\ &= \bigvee_{i \in I} (F(x, x')^L \circ F(x_i, x)^L)(t_i) \text{ by continuity} \\ &= \bigvee_{i \in I} F(x_i, x')^L(t_i) \\ &\leq t', \end{aligned}$$

which makes $(x, t) \leq (x', t')$. Hence ΣF is a cpo.

A routine argument shows ΣF is bounded complete. Let $W = \{(x_i, t_i) \mid i \in I\}$ be a set with upper bound (y, u) . Then because $x_i \leq y$ for all $i \in I$ there is a least upper bound $x = \bigvee_{i \in I} x_i$ in C . Because $F(x_i, y)^L(t_i) \leq u$ for all $i \in I$ we see $F(x_i, x)^L(t_i) = (F(x, y)^R \circ F(x_i, y)^L)(t_i) \leq F(x, y)^R(u)$ for all $i \in I$ in $F(x)$. Hence their least upper bound $t = \bigvee_{i \in I} F(x_i, x)^L(t_i)$ exists in $F(x)$. It follows that (x, t) is a least upper bound of W .

The cpo ΣF is also algebraic with finite elements of the form (e, f) where $e \in \mathbf{B}_C$ and $f \in \mathbf{B}_{F(e)}$. Such elements are certainly always finite by the following argument. Suppose $(e, f) \leq \bigvee V$ where V is a directed subset of ΣF , assumed to be of the form $V = \{(x_i, t_i) \mid i \in I\}$. As we have seen such a directed set V has least upper bound (x, t) where $x = \bigvee_{i \in I} x_i$ and t is the least upper bound of the directed set $\{F(x_i, x)^L(t_i) \mid i \in I\}$. Because $e \leq \bigvee_{i \in I} x_i$ and e is finite there is some $j \in I$ for which $e \leq x_j$. Because $F(e, x)^L(f) \leq \bigvee_{i \in I} F(x_i, x)^L(t_i)$ and $F(e, x)^L(f)$ is finite, being the image under an embedding of a finite element f , there is some $k \in I$ such that $F(e, x)^L(f) \leq F(x_k, x)^L(t_k)$ and $x_j \leq x_k$. From

$$F(x_k, x)^L \circ F(e, x_k)^L = F(e, x)^L,$$

we see $F(e, x_k)^L = F(x_k, x)^R \circ F(e, x)^L$. Hence $F(e, x_k)^L(f) \leq F(x_k, x)^R \circ F(x_k, x)^L(t_k) = t_k$ so $(e, f) \leq (x_k, t_k)$. Thus (e, f) is indeed finite.

Let $(x, t) \in \Sigma F$. Consider the set

$$V = \{(e, f) \leq (x, t) \mid e \in \mathbf{B}_C \text{ and } f \in \mathbf{B}_{F(e)}\}.$$

If $(e_0, f_0), (e_1, f_1) \in V$ then, as we saw when showing ΣF is bounded complete, their least upper bound has the form

$$(e_0 \vee e_1, F(e_0, e_0 \vee e_1)(f_0) \vee F(e_1, e_0 \vee e_1)(f_1))$$

, and this is an element of V using the fact that least upper bounds of finite elements are finite. Thus V is directed. From the fact that F is continuous we now show V has least upper bound (x, t) . Certainly, the set $\{e \leq x \mid e \in \mathbf{B}_C\}$ is directed with least upper bound x . We are assuming that Ff is continuous,

i.e. that it preserves directed colimits, so the colimiting cone $\{(e, x) \mid e \leq x \text{ and } e \in \mathbf{B}_C\}$ in C is sent to the colimiting cone $\{F(e, x) : F(e) \rightarrow F(x) \mid e \leq x \text{ and } e \in \mathbf{B}_C\}$ in \mathbf{D}^{EP} . By Theorem 2, this ensures

$$t = \bigvee \{F(e, x)^L \circ F(e, x)^R(t) \mid e \leq x \text{ and } e \in \mathbf{B}_C\}.$$

But now we see

$$t = \bigvee \{F(e, x)^L(f) \mid e \leq x \text{ and } e \in \mathbf{B}_C \text{ and } f \leq F(e, x)^R(t) \text{ and } f \in \mathbf{B}_{F(e)}\}.$$

This makes $(x, t) = \bigvee V$.

Now we can see directly that any finite element (x, t) must be such that $x \in \mathbf{B}_C$ and $t \in \mathbf{B}_{F(x)}$; because (x, t) is finite and the lub of a directed set of elements of this form it must be equal to one such element. And, of course, any element of ΣF is a least upper bound of finite elements. Clearly the set of finite elements is countable. This completes the proof that ΣF is a domain.

It is easy to see it comes equipped with a continuous projection function $p : \Sigma F \rightarrow C$. ■

Now we turn our attention to ΠF when F is a continuous functor $C \rightarrow \mathbf{D}^{\text{EP}}$ from a domain C . Its elements are continuous sections. A section is a functor $s : C \rightarrow \Sigma F$ such that $p \circ s = \text{id}_C$. Bearing in mind the nature of ΣF we take the image of $x \in C$ under s to be $s(x) = (x, t_x)$. As both categories C and ΣF are partial orders, s being a functor amounts to monotonicity, *i.e.*

$$\begin{aligned} x \leq y &\text{ implies } s(x) \leq s(y), \\ \text{i.e. } x \leq y &\text{ implies } (x, t_x) \leq (y, t_y), \\ \text{i.e. } x \leq y &\text{ implies } F(x, y)(t_x) \leq t_y \end{aligned} \tag{1}$$

for all $x, y \in C$. Sections thus correspond to families $\langle t_x \rangle_{x \in C}$ which satisfy (1). Continuous sections correspond to families which satisfy the monotonicity condition (1) and

$$t \bigvee_V = \bigvee_{v \in V} F(v, \bigvee V)(t_v) \tag{2}$$

for any directed set V of C . We call such families continuous. Two continuous sections s, s' correspond to continuous families $t = \langle t_x \rangle_{x \in C}$ and $t' = \langle t'_x \rangle_{x \in C}$ respectively. A morphism between them corresponds to a family of morphisms $\langle \alpha_x : t_x \rightarrow t'_x \rangle_{x \in C}$ but each such component α_x simply amounts to an ordering $t_x \leq t'_x$. Hence, a morphism $s \rightarrow s'$ between sections corresponds to a pointwise ordering

$$t \leq t' \text{ iff } \forall x \in C. t_x \leq t'_x$$

between the corresponding families.

Not surprisingly, to show ΠF is a domain it is convenient to work with the isomorphic category of continuous families with morphisms given by the pointwise order. Clearly this category is a partial order, and, as we now show, it is a domain.

Proposition 8 *Let C be a domain. Let $F : C \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor. Then ΠF is a domain.*

Proof: There is a least family with each component consisting of $\perp_{F(x)}$ for $x \in C$. Let $\{t^{(i)} \mid i \in I\}$ be a directed set in ΠF . Define the family $t = \langle \bigvee_{i \in I} t_x^{(i)} \rangle_{x \in C}$. Clearly it satisfies (1). Let V be a directed subset of C . Then

$$\begin{aligned} t \bigvee V &= \bigvee_{i \in I} t \bigvee V^{(i)} \\ &= \bigvee_{i \in I} \bigvee_{v \in V} F(v, \bigvee V)^L(t_v^{(i)}) \\ &= \bigvee_{v \in V} \bigvee_{i \in I} F(v, \bigvee V)^L(t_v^{(i)}), \\ &= \bigvee_{v \in V} F(v, \bigvee V)^L(\bigvee_{i \in I} t_v^{(i)}) \\ &= \bigvee_{v \in V} F(v, \bigvee V)^L(t_v) \end{aligned}$$

so t satisfies (2) and is therefore a continuous family. Thus ΠF is a cpo.

To show ΠF is bounded complete, assume $\{t^{(i)} \mid i \in I\}$, a set of continuous families, has upper bound u . As $F(x)$ is a domain and so bounded complete for all $x \in C$ we can define a family $t = \langle \bigvee_{i \in I} t_x^{(i)} \rangle_{x \in C}$. It satisfies (1) above. Let V be a directed subset of C . Then, to show (2), we notice

$$\begin{aligned} t \bigvee V &= \bigvee_{i \in I} t \bigvee V^{(i)} \\ &= \bigvee_{i \in I} \bigvee_{v \in V} F(v, \bigvee V)^L(t_v^{(i)}) \\ &= \bigvee_{v \in V} \bigvee_{i \in I} F(v, \bigvee V)^L(t_v^{(i)}), \\ &= \bigvee_{v \in V} F(v, \bigvee V)^L(\bigvee_{i \in I} t_v^{(i)}) \\ &= \bigvee_{v \in V} F(v, \bigvee V)^L(t_v) \end{aligned}$$

where we have used the fact that embeddings preserve least upper bounds.

Let $e \in \mathbf{B}_C$ and $f \in \mathbf{B}_{F(e)}$. Define the family $[e, f]$ to have component

$$[e, f]_x = \begin{cases} F(e, x)^L(f) & \text{if } e \leq x, \\ \perp_{F(x)} & \text{otherwise,} \end{cases}$$

for $x \in C$. It is easy to check $[e, f]$ satisfies (1) and (2) and so is a continuous family. Consider a family t , obtained in the following way as the least upper bound of a finite number of such families,

$$t = [e_1, f_1] \vee \cdots \vee [e_n, f_n].$$

We show t finite. Suppose $t \leq \bigvee V$ where V is a directed subset of C . Then for any i , with $1 \leq i \leq n$, we get

$$f_i \leq t_{e_i} \leq (\bigvee V)_{e_i} = \bigvee_{v \in V} v_{e_i},$$

the least upper bound of a directed set. As f_i is finite, $f_i \leq v_{e_i}^{(i)}$ for some $v^{(i)} \in V$. But then $[e_i, f_i] \leq v^{(i)}$. As V is directed there is some $v \in V$ which dominates each $v^{(i)}$ for $0 \leq i \leq n$ which ensures $t \leq v$. This shows t is finite.

A continuous family t is easily seen to be the least upper bound of the directed set

$$\{[e_1, f_1] \vee \cdots \vee [e_n, f_n] \mid f_1 \leq t_{e_1} \& \cdots \& f_n \leq t_{e_n}\},$$

where we are assured that the least upper bounds mentioned exist because they are bounded above in a bounded-complete partial order. It follows that any family which is a finite element of ΠF must have the form $[e_1, f_1] \vee \cdots \vee [e_n, f_n]$. Clearly such elements form a countable set. Hence ΠF is a domain. ■

3.3 Families indexed by a category of domains.

Our other important example arises when $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ is a continuous functor. In this case, as we shall see, while ΣF can only be considered as a category, ΠF is isomorphic to a domain when both are viewed as categories.

Assume $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ is a continuous functor. In this case, ΣF is a category with objects pairs (X, t_X) , where $X \in \mathbf{D}^{\text{EP}}$ and $t_X \in F(X)$, and morphisms $(X, t_X) \rightarrow (Y, t_Y)$ correspond to morphisms $f : X \rightarrow Y$ for which $(Ff)^L t_X \leq t_Y$. Note, ΣF is not a partial order—it simply has too many morphisms. We need to consider the form of colimits in ΣF . A directed family in ΣF corresponds to a directed set $\langle I, \leq \rangle$ indexing a family of objects (X_i, t_i) in ΣF and morphisms $f_{ij} \in \mathbf{D}^{\text{EP}}(X_i, X_j)$ so that $(Ff_{ij})^L t_i \leq t_j$, for $i \leq j$. A colimit for such a family corresponds to a pair (X, t) with a collection of morphisms $\langle g_i : X_i \rightarrow X \rangle_{i \in I}$ making a colimiting cone in \mathbf{D}^{EP} and so that $t = \bigvee_i (Fg_i)^L t_i$.

As in the earlier case, when $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ the category ΠF of continuous sections can be seen as consisting of certain kinds of continuous families ordered pointwise. As before, sections correspond to families $\langle t_X \rangle_{X \in \mathbf{D}^{\text{EP}}}$, where $t_X \in F(X)$, which are monotonic in that they satisfy

$$f \in \mathbf{D}^{\text{EP}}(X, Y) \text{ implies } (Ff)^L t_X \leq t_Y \quad (1)$$

for any f . Continuous sections preserve directed colimits. Thus if $\langle \rho_i : X_i \rightarrow X \rangle_{i \in I}$ is a directed colimit in \mathbf{D}^{EP} , then $\langle s\rho_i : sX_i \rightarrow sX \rangle_{i \in I}$ is a directed colimit in ΣF . Considering the form of directed colimits in ΣF , it follows that continuous sections correspond to families which satisfy (1) and also the requirement that for such directed colimits $\langle \rho_i : X_i \rightarrow X \rangle_{i \in I}$ in \mathbf{D}^{EP} we have $t_X = \bigvee_i (F\rho_i)^L t_{X_i}$. Recalling Theorem 2 we can write this condition as follows. For any cone $\langle \rho_i : X_i \rightarrow X \rangle_{i \in I}$ we have

$$\{\rho_i^L \circ \rho_i^R \mid i \in I\} \text{ is directed and } \bigvee_{i \in I} \rho_i^L \circ \rho_i^R = \text{id}_X \text{ implies } t_X = \bigvee_i (F\rho_i)^L t_{X_i}. \quad (2)$$

We call families satisfying (1) and (2) continuous. As before, morphisms between continuous sections correspond to their associated families being ordered pointwise, *i.e.*

$$t \leq t' \text{ iff } \forall X \in \mathbf{D}^{\text{EP}}. t_X \leq t'_X$$

where t and t' are two continuous families.

At this point it is tempting to conclude that ΠF is a partially ordered set and press on with the demonstration that it is a domain. Unfortunately, it is not quite, as its objects, the continuous sections, are not sets. Even though the elements of ΠF are classes they can be put in 1-1 correspondence with the

elements of a suitable set. To see this, take \mathbf{S} to be some countable subcategory of domains equivalent to the full subcategory of all finite domains with embedding-projection pairs as morphisms. Then any continuous section is determined by its restriction to the standard domains \mathbf{S} . Ordered pointwise these restrictions are in 1-1 order preserving correspondence with ΠF . In this sense ΠF is isomorphic to a partially ordered set, in fact a domain. This more generous sense of isomorphism is quite standard in category theory; according to the usual notion of isomorphism there, ΠF is isomorphic to a domain when both are viewed as categories. This has described the sense in which we mean ΠF is isomorphic to a domain. Details are given in the proof of the following theorem.

Theorem 9 *Let $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor. The category ΠF is isomorphic to a domain.*

Proof: Take $\Pi_{\mathbf{S}} F$ to be the partial order consisting of families $\langle t_X \rangle_{X \in \mathbf{S}}$ which are monotonic in the sense that

$$f \in \mathbf{D}^{\text{EP}}(X, Y) \text{ implies } (Ff)^L t_X \leq t_Y,$$

for all $X, Y \in \mathbf{S}$, ordered pointwise. It is clear that $\Pi_{\mathbf{S}} F$ is a set because \mathbf{S} is. Now we show that ΠF and $\Pi_{\mathbf{S}} F$ are isomorphic as categories, and, later, that $\Pi_{\mathbf{S}} F$ is a domain.

Clearly, any continuous section $t \in \Pi F$ determines, by restriction, an element $\text{res } t \in \Pi_{\mathbf{S}} F$. Conversely, any element of $t \in \Pi_{\mathbf{S}} F$ can be extended to a continuous section $\text{ext } t$ by taking

$$(\text{ext } t)_D = \bigvee \{(Ff)^L t_X \mid X \in \mathbf{S} \text{ \& } f \in \mathbf{D}^{\text{EP}}(X, D)\},$$

for any domain D . This must be checked to be well-defined however.

We note the set $\{(Ff)^L t_X \mid X \in \mathbf{S} \text{ \& } f \in \mathbf{D}^{\text{EP}}(X, D)\}$ is directed so that the least upper bound really does exist. To show this, take two elements of the set $y_0 = (Ff_0)^L t_{X_0}$ and $y_1 = (Ff_1)^L t_{X_1}$ arising from morphisms $f_0 \in \mathbf{D}^{\text{EP}}(X_0, D)$ and $f_1 \in \mathbf{D}^{\text{EP}}(X_1, D)$ where X_0, X_1 are finite domains. By Lemma 4 there is a finite domain X and $g \in \mathbf{D}^{\text{EP}}(X, D)$, $g_0 \in \mathbf{D}^{\text{EP}}(X_0, X)$ and $g_1 \in \mathbf{D}^{\text{EP}}(X_1, X)$ with $f_0 = g \circ g_0$ and $f_1 = g \circ g_1$. Because t is monotonic it follows that $y_0, y_1 \leq (Fg)^L t_X$, an element of the set. Hence the set is directed, and the definition above does at least yield a family. It remains to show that the family is continuous. Firstly, to show the family is monotonic, assume $g \in \mathbf{D}^{\text{EP}}(D, E)$ and notice

$$\begin{aligned} (Fg)^L(\text{ext } t)_D &= (Fg)^L \bigvee \{(Ff)^L t_X \mid X \in \mathbf{S} \text{ \& } f \in \mathbf{D}^{\text{EP}}(X, D)\} \\ &= \bigvee \{(Fg)^L \circ (Ff)^L t_X \mid X \in \mathbf{S} \text{ \& } f \in \mathbf{D}^{\text{EP}}(X, D)\} \\ &= \bigvee \{(F(g \circ f))^L t_X \mid X \in \mathbf{S} \text{ \& } f \in \mathbf{D}^{\text{EP}}(X, D)\} \\ &\leq \bigvee \{(Fh)^L t_X \mid X \in \mathbf{S} \text{ \& } h \in \mathbf{D}^{\text{EP}}(X, E)\} \\ &= (\text{ext } t)_E. \end{aligned}$$

This shows monotonicity. Suppose now that $\langle \rho_i \in \mathbf{D}^{\text{EP}}(D_i, D) \rangle_{i \in I}$ is a directed colimit. To complete the demonstration of continuity we require that

$$(\text{ext } t)_D = \bigvee \{(F\rho_i)^L(\text{ext } t_{D_i}) \mid i \in I\}.$$

Note first that the set is directed because $\text{ext } t$ is monotonic. Again by monotonicity we obtain

$$(\text{ext } t)_D \geq \bigvee \{(F\rho_i)^L(\text{ext } t_{D_i}) \mid i \in I\}.$$

According to its definition $(\text{ext } t)_D$ is the least upper bound of elements $(Ff)^L t_X$ for $X \in \mathbf{S}$ and $f \in \mathbf{D}^{\text{EP}}(X, D)$. Consider such an element. By Lemma 5, there is some $i \in I$ and $h \in \mathbf{D}^{\text{EP}}(X, D_i)$ such that $f = \rho_i \circ h$. Now we see

$$(Ff)^L t_X = (F(\rho_i \circ h))^L t_X = (F\rho_i)^L((Fh)^L t_X) \leq (F\rho_i)^L(\text{ext } t_{D_i}).$$

It follows that $(\text{ext } t)_D \leq \bigvee_i (F\rho_i)^L(\text{ext } t_{D_i})$, and now the equality required for continuity is obvious.

Now, it is easy to see that the two operations restriction $\text{res} : \Pi F \rightarrow \Pi_S F$ and extension $\text{ext} : \Pi_S F \rightarrow \Pi F$ preserve the order relation. For $t \in \Pi_S F$, we certainly have $t_Y \leq (\text{ext } t)_Y$ for $Y \in \mathbf{S}$ —consider the identity morphism on Y —and by the monotonicity of t we see

$$(\text{res } \text{ext } t)_Y = \bigvee \{(Ff)^L t_X \mid X \in \mathbf{S} \text{ and } f \in \mathbf{D}^{\text{EP}}(X, Y)\} \leq t_Y.$$

Hence $\text{res } \text{ext } t = t$ for $t \in \Pi_S F$. For $X \in \mathbf{S}$ we have $(\text{res } t)_X = t_X$, so from the definition of ext and res we see

$$(\text{ext } \text{res } t)_D = \bigvee \{(Ff)^L t_X \mid X \in \mathbf{S} \text{ and } f \in \mathbf{D}^{\text{EP}}(X, D)\},$$

for a domain D . However, because t is continuous and D is the colimit of finite embeddings in the sense of Theorem 3, we also have

$$t_D = \bigvee \{(Ff)^L t_X \mid X \in \mathbf{S} \text{ and } f \in \mathbf{D}^{\text{EP}}(X, D)\}.$$

Hence $\text{ext } \text{res } t = t$, for all $t \in \Pi F$. We conclude that $\text{res} : \Pi F \rightarrow \Pi_S F$ and $\text{ext} : \Pi_S F \rightarrow \Pi F$ form an order isomorphism.

We now show $\Pi_S F$ is a domain. It has a least element, the family $\langle \perp_X \rangle_{X \in \mathbf{S}}$. Suppose $\{t^{(i)} \mid i \in I\}$ is a directed set in $\Pi_S F$. Define the family t by taking

$$t_X = \bigvee_i t_X^{(i)},$$

for all $X \in \mathbf{S}$ —the least upper bound exists because the set $\{t_X^{(i)} \mid i \in I\}$ is directed because $\{t^{(i)} \mid i \in I\}$ is. It is monotonic because, supposing $f \in \mathbf{D}^{\text{EP}}(X, Y)$, we see

$$(Ff)^L(t_X) = (Ff)^L\left(\bigvee_i t_X^{(i)}\right) = \bigvee_i (Ff)^L(t_X^{(i)}) \leq \bigvee_i t_Y^{(i)},$$

using the fact that $(Ff)^L$ is continuous. A very similar argument shows that $\Pi_S F$ is bounded complete though in this case the argument uses the fact that embeddings preserve all existing least upper bounds.

Suppose there is a monotone family t such that $t_X = e \in FX$ is finite for some $X \in \mathbf{S}$. Define

$$[X, e]_Y = \bigvee \{(Ff)^L(e) \mid f : X \rightarrow Y\}$$

This is well-defined since t_Y is a bound for the set whose join is being taken on the right. It is possible to show that it is a monotone family which does not depend on the choice of t . Now, any least upper bound which exists of the form

$$[X_1, e_1] \vee \cdots \vee [X_n, e_n],$$

where $e_1 \in FX_1, \dots, e_n \in FX_n$, is a finite element of $\Pi_S F$. The remaining argument, showing that any element of $\Pi_S F$ is the lub of such elements and that all finite elements have this form, echoes that in the proof of Proposition 8, and we omit it. Having chosen S to be countable it follows that the finite elements form a countable set, and hence that $\Pi_S F$ is a domain isomorphic to ΠF . ■

Thus although strictly speaking the category ΠF is not a partial order because its objects are classes, not sets, it is nevertheless isomorphic to a domain. Because of this, in the future, we shall treat ΠF as a domain, in fact as the domain with continuous families as elements, and not fuss about this problem with foundations. The more fastidious reader can after all replace our construction of ΠF with the isomorphic small category $\Pi_S F$ provided in the proof above.

3.4 Π with parameters.

In the discussion later we will often need to use the Π operator with parameters. If $F : \mathbf{C} \times \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ is continuous, then we write $\Pi^{\mathbf{C}} F : \mathbf{C} \rightarrow \mathbf{D}^{\text{EP}}$ for the continuous functor defined as follows. The action of $\Pi^{\mathbf{C}} F$ on objects is given by $(\Pi^{\mathbf{C}} F)(C) = \Pi(F(C, -))$. Given $f \in \mathbf{C}(C, D)$, we define

$$(\Pi^{\mathbf{C}} F)(f) \in \mathbf{D}^{\text{EP}}((\Pi^{\mathbf{C}} F)(C), (\Pi^{\mathbf{C}} F)(D))$$

by taking

$$\begin{aligned} (\Pi^{\mathbf{C}} F)^L(f)(s)_Z &= F^L(f, \text{id}_Z)(s_Z) \\ (\Pi^{\mathbf{C}} F)^R(f)(t)_Z &= F^R(f, \text{id}_Z)(t_Z) \end{aligned}$$

for each section $s \in (\Pi^{\mathbf{C}} F)(C)$ and $t \in (\Pi^{\mathbf{C}} F)(D)$.

Of course, we must show that this definition makes sense. First of all, let us check that $(\Pi^{\mathbf{C}} F)^L(f)(s) \in (\Pi^{\mathbf{C}} F)(D)$. Suppose $s \in (\Pi^{\mathbf{C}} F)(C) = \Pi(F(C, -))$ and let $t_X = (\Pi^{\mathbf{C}} F)^L(f)(s)_X = F^L(f, \text{id}_X)(s_X)$, we wish to show that $t_X \in \Pi(F(D, -))$. Suppose $g \in \mathbf{D}^{\text{EP}}(X, Y)$. Then

$$\begin{aligned} F^L(\text{id}_D, g)(t_X) &= F^L(\text{id}_D, g)(F^L(f, \text{id}_X)(s_X)) \\ &= F^L(f, \text{id}_X)(F^L(\text{id}_D, g)(s_X)) \\ &\leq F^L(f, \text{id}_X)((s_Y)) \\ &= t_Y. \end{aligned}$$

This proves monotonicity. To prove continuity, suppose $g_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions $g_i^L \circ g_i^R$ form

a directed collection such that $\bigvee_i g_i^L \circ g_i^R = \text{id}_X$, then

$$\begin{aligned}
\bigvee_i F^L(\text{id}_D, g_i)(t_{X_i}) &= \bigvee_i F^L(\text{id}_D, g_i)(F^L(f, \text{id}_{X_i})(s_{X_i})) \\
&= \bigvee_i F^L(f, \text{id}_X)(F^L(\text{id}_D, g_i)(s_{X_i})) \\
&= F^L(f, \text{id}_X)(\bigvee_i F^L(\text{id}_D, g_i)(s_{X_i})) \\
&= F^L(f, \text{id}_X)(s_X) \\
&= t_X
\end{aligned}$$

so $(\Pi^{\mathbf{C}} F)^L(f)(s) \in (\Pi^{\mathbf{C}} F)(D)$.

Now suppose $t \in (\Pi^{\mathbf{C}} F)(D) = \Pi(F(D, -))$ and let $s_X = (\Pi^{\mathbf{C}} F)^R(f)(t)_X = F^R(f, \text{id}_X)(t_X)$. We wish to show that $s \in (\Pi^{\mathbf{C}} F)(C) = \Pi(F(D, -))$. Suppose $g \in \mathbf{D}^{\text{EP}}(X, Y)$. Then

$$\begin{aligned}
s_X &= F^R(f, \text{id})(t_X) \\
&\leq F^R(f, \text{id})(F^R(\text{id}, g)(t_Y)) \\
&= F^R(\text{id}, g)(F^R(f, \text{id})(t_Y)) \\
&= F^R(\text{id}, g)(s_Y)
\end{aligned}$$

This proves monotonicity. To prove continuity, suppose $g_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions $g_i^L \circ g_i^R$ form a directed collection such that $\bigvee_i g_i^L \circ g_i^R = \text{id}_X$. To keep the notation simple, let

$$\begin{aligned}
\phi_i &= F(f, \text{id}_{X_i}) \in \mathbf{D}^{\text{EP}}(F(C, X_i), F(D, X_i)) \\
\alpha_i &= F(\text{id}_D, g_i) \in \mathbf{D}^{\text{EP}}(F(D, X_i), F(D, X)) \\
\beta_i &= F(\text{id}_C, g_i) \in \mathbf{D}^{\text{EP}}(F(C, X_i), F(C, X)) \\
\phi &= F(f, \text{id}_X) \in \mathbf{D}^{\text{EP}}(F(C, X), F(D, X))
\end{aligned}$$

Note that

$$\begin{aligned}
\beta_i^R \circ \phi^R \circ \alpha_i^L &= F^R(\text{id}_C, g_i) \circ F^R(f, \text{id}_X) \circ F^L(\text{id}_D, g_i) \\
&= F^R(f, \text{id}_{X_i}) \circ F^R(\text{id}_D, g_i) \circ F^L(\text{id}_D, g_i) \\
&= F^R(f, \text{id}_{X_i}) \\
&= \phi_i^R.
\end{aligned}$$

Since $\bigvee_i \alpha_i^L \circ \alpha_i^R = \text{id}_{F(D, X)}$ and $\bigvee_i \beta_i^L \circ \beta_i^R = \text{id}_{F(C, X)}$, we have

$$\begin{aligned}
\phi^R &= (\bigvee_i \beta_i^L \circ \beta_i^R) \circ \phi^R \circ (\bigvee_i \alpha_i^L \circ \alpha_i^R) \\
&= \bigvee_i \beta_i^L \circ (\beta_i^R \circ \phi^R \circ \alpha_i^L) \circ \alpha_i^R \\
&= \bigvee_i \beta_i^L \circ (\phi_i^R) \circ \alpha_i^R
\end{aligned}$$

Now, let $s_X = (\Pi^C F)^R(f)(t)_X = \phi^R(t_X)$ and $s_{X_i} = (\Pi^C F)^R(f)(t)_{X_i} = \phi^R(t_{X_i})$. Then

$$\begin{aligned}
s_X &= \phi^R(t_X) \\
&= (\bigvee_i \beta_i^L \circ (\phi_i^R) \circ \alpha_i^R)(\bigvee_i \alpha_i^L(t_{X_i})) \\
&= \bigvee_i (\beta_i^L \circ (\phi_i^R))(t_{X_i}) \\
&= \bigvee_i \beta_i^L(s_{X_i}).
\end{aligned}$$

That is, $s_X = \bigvee_i F^L(\text{id}_C, g_i)(s_{X_i})$ and therefore $s \in (\Pi^C F)(C) = \Pi(F(D, -))$.

We have now shown that the definitions of $(\Pi^C F)^L(f)$ and $(\Pi^C F)^R(f)$ make sense. The proof that $(\Pi^C F)(f) \in \mathbf{D}^{\text{EP}}((\Pi^C F)(C), (\Pi^C F)(D))$ and the proof that $\Pi^C F$ is a continuous functor are both routine.

Notation: Later we shall be concerned with functors $F : \mathbf{C} \times \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ and the associated Π^C in the case where $\mathbf{C} = (\mathbf{D}^{\text{EP}})^m$. In this case we shall write Π^m for Π^C .

3.5 Information systems.

The inspiration for our work came originally from Girard's paper [8]. There he uses a representation of qualitative domains with morphisms stable functions and rigid embeddings to give a model for the second-order λ -calculus. For domains, we can use the representation of information systems in a similar way to give an interesting, elementary construction of ΠF for a functor F on domains. We give a sketch of the approach based on the presentation of information systems in [12] following [23]. Because the proofs are straightforward and not essential for what follows we omit them.

Recall the definition of an information system:

Definition: An *information system* is defined to be a structure (A, Con, \vdash) , where A is a countable set (the *tokens*), Con is a non-null subset of finite subsets of A (the *consistent* sets) and \vdash is a subset of $\text{Con} \times A$ (the *entailment relation*) which satisfy:

- $X \subseteq Y \in \text{Con}$ implies $X \in \text{Con}$
- $a \in A$ implies $\{a\} \in \text{Con}$
- $X \vdash a$ implies $X \cup \{a\} \in \text{Con}$
- $X \in \text{Con}$ and $a \in X$ implies $X \vdash a$
- $(X, Y \in \text{Con} \text{ and } \forall b \in Y. X \vdash b \text{ and } Y \vdash c)$ implies $X \vdash c$.

An information system determines a domain:

Proposition 10 *The elements of an information system (A, Con, \vdash) are defined to be those subsets x of A which satisfy:*

- $X \subseteq x$ implies $X \in \text{Con}$ for any finite set X ,
- $X \subseteq x$ and $X \vdash a$ implies $a \in x$.

Ordering the elements by inclusion we obtain a domain $|A|$ with finite elements precisely the sets $\{a \in A \mid \exists X \subset Y. X \vdash a\}$, obtained from $X \in \text{Con}$.

A domain determines an information system:

Definition: Let D be a domain. Define $ID = (\mathbf{B}_D, \text{Con}, \vdash)$ where \mathbf{B}_D is the set of finite elements of D and Con and \vdash are defined as follows:

$X \in \text{Con}$ iff $X \subset \mathbf{B}_D$ and X is finite and X is bounded,

$X \vdash e$ iff $X \in \text{Con}$ and $e \leq \bigvee X$.

Proposition 11 Let D be a domain. Then ID is an information system with domain of elements $|ID|$ isomorphic to D . The isomorphism pair is

$\theta : D \rightarrow |ID|$ given by $\theta : d \mapsto \{e \in \mathbf{B}_D \mid e \leq d\}$,

$\phi : |ID| \rightarrow D$ given by $\phi : x \mapsto \bigvee x$.

As is well-known a continuous function f between domains is determined by its action on finite elements and so by the relation f^0 between finite elements that it induces, a relation defined as follows.

Definition: Let $f : D \rightarrow E$ be a continuous function between domains. Define $f^0 = \{(d, e) \in \mathbf{B}_D \times \mathbf{B}_E \mid e \leq f(d)\}$.

Embeddings between domains correspond to the following kinds of mappings between the finite elements of the associated information systems.

Proposition 12 Let $f : D \rightarrow E$ be a continuous function between domains D and E . The function f is an embedding iff

- f^0 is a 1-1 function $\mathbf{B}_D \rightarrow \mathbf{B}_E$,
- $X \in \text{Con}_D$ iff $fX \in \text{Con}_E$, for all finite subsets X of \mathbf{B}_D , and
- $X \vdash_D d$ iff $fX \vdash_E f(d)$, for all elements d and finite subsets X of \mathbf{B}_D .

To define the information system of ΠF of a continuous functor on domains, as earlier, we use \mathbf{S} , a countable category equivalent to the full subcategory of finite domains with embedding-projection pairs.

Definition: Let $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor on domains. Take T^+ to consist of those pairs (X, b) where $X \in \mathbf{S}$ and $b \in \mathbf{B}_{F(X)}$. For W , a finite subset of T^+ , define

$W \in \text{Con}$ iff $\forall Y \in \mathbf{S}. \{(Ff)^L b \mid \exists X. (X, b) \in W \text{ and } f \in \mathbf{D}^{\text{EP}}(X, Y)\} \in \text{Con}_{FY}$.

Define the tokens T to be those elements (X, b) of T^+ for which $\{(X, b)\} \in \text{Con}$. For $W \in \text{Con}$ and $(Y, c) \in T$, define

$W \vdash (Y, c)$ iff $\{(Ff)^L b \mid \exists X. (X, b) \in W \text{ and } f \in \mathbf{D}^{\text{EP}}(X, Y)\} \vdash_{FY} c$.

Finally, define $\Pi_I F$ to be (C, Con, \vdash) .

Theorem 13 Let $F : \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ be a continuous functor on domains. Then

(i) $\Pi_I F$ is an information system.

(ii) $\Pi F \cong |\Pi_I F|$ with isomorphism pair $\theta : \Pi F \rightarrow |\Pi_I F|$ and $\phi : |\Pi_I F| \rightarrow \Pi F$ given by

$$\begin{aligned}\theta(t) &= \{(X, b) \mid b \leq t_X \text{ and } b \in \mathbf{B}_{F(X)}\}, \\ \phi(x) &= \langle t_Y \rangle_{Y \in \mathbf{D}^{\text{EP}}} \text{ where} \\ t_Y &= \{(Ff)^L b \mid \exists X. f : X \rightarrow Y \text{ and } (X, b) \in x\}. \quad \blacksquare\end{aligned}$$

4 Basic combinators.

Here we introduce the notation and results we shall use to provide a semantics for the polymorphic λ -calculus. We are concerned with functors on the category \mathbf{D}^{EP} . Suppose F_1, \dots, F_n are continuous functors from $(\mathbf{D}^{\text{EP}})^m$ into \mathbf{D}^{EP} . We claim that \mathbf{p}^i , the projection map defined earlier, is a section of $\#(F_1, \dots, F_n) \Rightarrow F_i$. To check this, suppose $f \in (\mathbf{D}^{\text{EP}})^m(X, Y)$. Then

$$\begin{aligned}(\#(F_1, \dots, F_n) \Rightarrow F_i)^R(f)(\mathbf{p}_Y^i)(x_1, \dots, x_n) \\ &= (F_i^R(f) \circ \mathbf{p}_Y^i \circ \#(F_1, \dots, F_n)^L(f))(x_1, \dots, x_n) \\ &= F_i^R(f)(F_i^L(f)(x_i)) \\ &= \mathbf{p}_X^i(x_1, \dots, x_n).\end{aligned}$$

It is clear that \mathbf{p}^i will be a continuous section.

Let $P, F, G : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$ be continuous functors. Suppose s is a continuous section of the functor $P \Rightarrow (F \Rightarrow G) : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$ and t is a continuous section of the functor $P \Rightarrow F : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$. We define a continuous section $\text{apply}(s, t)$ of $P \Rightarrow G$ by the equation

$$\text{apply}(s, t)_X(x) = (s_X(x))(t_X(x))$$

where $x \in P(X)$. To show that $\text{apply}(s, t)$ really is a section, suppose $f \in (\mathbf{D}^{\text{EP}})^m(X, Y)$. Then

$$\begin{aligned}(P \Rightarrow G)^L(f)(\text{apply}(s, t)_X(x)) \\ &= (G^L(f) \circ (\text{apply}(s, t)_X) \circ P^R(f))(x) \\ &= G^L(f)((\text{apply}(s, t)_X)(P^R(f)(x))) \\ &= G^L(f)((s_X(P^R(f)(x)))(t_X(P^R(f)(x)))) \\ &\leq G^L(f)((F \Rightarrow G)^R(f)(s_Y(x))(F^R(f)(t_Y(x)))) \\ &= G^L(f)((G^R(f) \circ (s_Y(x)) \circ F^L(f))(F^R(f)(t_Y(x)))) \\ &\leq (s_Y(x))(t_Y(x)) \\ &= \text{apply}(s, t)_Y(x).\end{aligned}$$

To see that $\text{apply}(s, t)$ is continuous, suppose $f_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions $f_i^L \circ f_i^R$ form a directed collection such that $\bigvee_i f_i^L \circ f_i^R = \text{id}_X$, then

$$\bigvee_i (P \Rightarrow G)^L(f_i)(\text{apply}(s, t)_{X_i})(x)$$

$$\begin{aligned}
&= \bigvee_i G^L(f_i)((s_{X_i}(P^R(f_i)(x)))(t_{X_i}(P^R(f_i)(x)))) \\
&= \bigvee_i G^L(f_i)((F \Rightarrow G)^R(f_i)((P \Rightarrow (F \Rightarrow G))^L(f_i)(s_{X_i})(x)))(F^R(f_i)((P \Rightarrow F)^L(f_i)(t_{X_i})(x))) \\
&= \bigvee_i G^L(f_i)((F \Rightarrow G)^R(f_i)(s_X(x)))(F^R(f_i)(t_X(x))) \\
&= \bigvee_i ((G^L(f_i) \circ G^R(f_i))(s_X(x))((F^L(f_i) \circ F^R(f_i))(t_X(x))) \\
&= \text{apply}(s, t)_X(x).
\end{aligned}$$

Let

$$\begin{aligned}
P &: (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}, \\
F &: (\mathbf{D}^{\text{EP}})^m \times \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}, \text{ and} \\
G &: (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}
\end{aligned}$$

be continuous functors. Suppose t is a continuous section of the functor

$$P \Rightarrow \Pi^m F : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}.$$

We define a continuous section $\text{Apply}(t, G)$ of the functor

$$P \Rightarrow (F \circ \langle \text{id}_{(\mathbf{D}^{\text{EP}})^m}, G \rangle) : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$$

by the equation

$$\text{Apply}(t, G)_X(x) = t_X(x)_{G(X)}$$

where $x \in P(X)$. We check that $\text{Apply}(t, G)$ is indeed a section; suppose $f \in (\mathbf{D}^{\text{EP}})^m(X, Y)$, then

$$\begin{aligned}
&(P \Rightarrow (F \circ \langle \text{id}_{(\mathbf{D}^{\text{EP}})^m}, G \rangle))^L(f)(\text{Apply}(t, G)_X) \\
&= (F^L(f, G(f)) \circ \text{Apply}(t, G)_X \circ P^R(f))(x) \\
&= F^L(f, G(f))((t_X \circ P^R(f))(x)_{G(X)}) \\
&= F^L(f, G(f))(((\Pi^m F)^R(f) \circ (\Pi^m F)^L(f) \circ t_X \circ P^R(f))(x)_{G(X)}) \\
&= F^L(f, G(f))(((\Pi^m F)^R(f) \circ (P \Rightarrow \Pi^m F)^L(f)(t_X))(x)_{G(X)}) \\
&\leq F^L(f, G(f))(((\Pi^m F)^R(f) \circ t_Y)(x)_{G(X)}) \\
&= F^L(f, G(f))((\Pi^m F)^R(f)(t_Y(x))_{G(X)}) \\
&= F^L(f, G(f))(F^R(f, \text{id}_{G(X)})(t_Y(x)_{G(X)})) \\
&= (F^L(\text{id}_Y, G(f)) \circ F^L(f, \text{id}_{G(X)}) \circ F^R(f, \text{id}_{G(X)}))(t_Y(x)_{G(X)}) \\
&\leq F^L(\text{id}_Y, G(f))(t_Y(x)_{G(X)}) \\
&\leq (t_Y(x))_{G(Y)} \\
&= \text{Apply}(t, G)_Y.
\end{aligned}$$

where the penultimate step follows from the fact that $t_Y(x)$ is a section of $F(Y, -)$ and $G(f) \in \mathbf{D}^{\text{EP}}(G(X), G(Y))$. To see that $\text{Apply}(t, G)$ is continuous, suppose $f_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions

$f_i^L \circ f_i^R$ form a directed collection such that $\bigvee_i f_i^L \circ f_i^R = \text{id}_X$, then

$$\begin{aligned}
& \bigvee_i (P \Rightarrow (F \circ (\text{id}_{\mathbf{D}^{\text{EP}}})^m, G))^L(f_i)(\text{Apply}(t, G)_{X_i}) \\
&= \bigvee_i F^L(f_i, G(f_i))(((\Pi^m F)^R(f_i) \circ (P \Rightarrow \Pi^m F)^L(f_i)(t_{X_i}))(x)_{G(X_i)}) \\
&= \bigvee_i F^L(f_i, G(f_i))(((\Pi^m F)^R(f_i) \circ t_X)(x)_{G(X_i)}) \\
&= \bigvee_i (F^L(\text{id}_X, G(f_i)) \circ F^L(f_i, \text{id}_{G(X_i)}) \circ F^R(f_i, \text{id}_{G(X_i)}))(t_X(x)_{G(X_i)}) \\
&= \bigvee_i F^L(\text{id}_X, G(f_i))(t_X(x)_{G(X_i)}) \\
&= (t_X(x))_{G(X)} \\
&= \text{Apply}(t, G)_Y.
\end{aligned}$$

Let $P, F, G : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$ be continuous functors and suppose t is a continuous section of the functor $P \# F \Rightarrow G : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$. Then we define a continuous section $\text{curry}(t)$ of the functor $P \Rightarrow (F \Rightarrow G)$ by setting

$$\text{curry}(t)_X(x)(y) = t_X(x, y)$$

for $x \in P(X)$ and $y \in F(X)$. To see that this does define a section, suppose $f \in (\mathbf{D}^{\text{EP}})^m(X, Y)$. Then

$$\begin{aligned}
& (P \Rightarrow (F \Rightarrow G))^L(f)(\text{curry}(t)_X(x)(y)) \\
&= ((F \Rightarrow G)^L(f) \circ (\text{curry}(t)_X) \circ P^R(f))(x)(y) \\
&= (G^L(f) \circ ((\text{curry}(t)_X)(P^R(f)(x))) \circ F^R(f))(y) \\
&= G^L(f)(t_X(P^R(f)(x), F^R(f)(y))) \\
&= ((P \# F \Rightarrow G)^L(f)(t_X))(x, y) \\
&\leq t_Y(x, y) \\
&= \text{curry}(t)_Y(x)(y).
\end{aligned}$$

To see that $\text{curry}(t)$ is continuous, suppose $f_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions $f_i^L \circ f_i^R$ form a directed collection such that $\bigvee_i f_i^L \circ f_i^R = \text{id}_X$, then

$$\begin{aligned}
& \bigvee_i (P \Rightarrow (F \Rightarrow G))^L(f_i)(\text{curry}(t)_{X_i})(x)(y) \\
&= \bigvee_i ((P \# F \Rightarrow G)^L(f_i)(t_{X_i}))(x, y) \\
&= t_X(x, y) \\
&= \text{curry}(t)_X(x)(y).
\end{aligned}$$

Let $P : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$, $F : (\mathbf{D}^{\text{EP}})^m \times \mathbf{D}^{\text{EP}} \rightarrow \mathbf{D}^{\text{EP}}$ and suppose t is a continuous section of $(P \circ \text{Fst}) \Rightarrow F$. Let $X \in (\mathbf{D}^{\text{EP}})^m$ and $x \in P(X)$. We define $\text{Curry}(t)_X(x)$ to be the continuous section of $F(X, -)$ given by the equation

$$\text{Curry}(t)_X(x)_Z = t_{(X, Z)}(x).$$

This makes sense because $t_{(X,Z)}$ is a continuous functor in Z . We wish to show that $\text{Curry}(t)$ is a section of $P \Rightarrow \Pi^m F$. In other words, we want to show that

$$(P \Rightarrow \Pi^m F)^L(f)(\text{Curry}(t)_X) \leq \text{Curry}(t)_Y$$

where $f \in (\mathbf{D}^{\text{EP}})^m(X, Y)$. Let $x \in P(X)$ and suppose $Z \in \mathbf{D}^{\text{EP}}$. Then

$$\begin{aligned} & (P \Rightarrow \Pi^m F)^L(f)(\text{Curry}(t)_X)(x)_Z \\ &= ((\Pi^m F)^L(f) \circ \text{Curry}(t)_X \circ P^R(f))(x)_Z \\ &= (\Pi^m F)^L(f)(\text{Curry}(t)_X(P^R(f)(x)))_Z \\ &= F^L(f, \text{id}_Z)(\text{Curry}(t)_X(P^R(f)(x)))_Z \\ &= F^L(f, \text{id}_Z)(t_{(X,Z)}(P^R(f)(x))) \\ &= F^L(f, \text{id}_Z)(t_{(X,Z)}((P \circ \text{Fst})^R(f, \text{id}_Z)(x))) \\ &= ((P \circ \text{Fst}) \Rightarrow F)^L(f, \text{id}_Z)(t_{(X,Z)})(x) \\ &\leq t_{(Y,Z)}(x) \\ &= (\text{Curry}(t)_Y(x))_Z. \end{aligned}$$

To see that $\text{Curry}(t)$ is continuous, suppose $f_i \in \mathbf{D}^{\text{EP}}(X_i, X)$ and the functions $f_i^L \circ f_i^R$ form a directed collection such that $\bigvee_i f_i^L \circ f_i^R = \text{id}_X$, then

$$\begin{aligned} & \bigvee_i (P \Rightarrow \Pi^m F)^L(f_i)(\text{Curry}(t)_{X_i})(x)_Z \\ &= \bigvee_i ((P \circ \text{Fst}) \Rightarrow F)^L(f_i, \text{id}_Z)(t_{(X_i,Z)})(x) \\ &= t_{(X,Z)}(x) \\ &= (\text{Curry}(t)_X(x))_Z. \end{aligned}$$

Notation: Suppose

$$\begin{aligned} P &: (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}, \\ F &: (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}, \text{ and} \\ G &: (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}} \end{aligned}$$

are functors. Given continuous sections

$$\begin{aligned} s &\in \Pi(P \# F \Rightarrow G) \\ t &\in \Pi(P \Rightarrow F), \end{aligned}$$

we define a continuous section

$$[t]s \in \Pi(P \Rightarrow G)$$

by setting

$$([t]s)_X(x) = \text{apply}(\text{curry}(s), t) = s_X(x, t_X(x)). \quad \blacksquare$$

We will need the following Lemma later:

Lemma 14 1. If $t'_X(p, b) = t_X(p)$ and $s'_X(p, b, a) = s_X(p, a, b)$ for every X , p , a and b , then $\text{curry}([t']s') = [t](\text{curry}(s))$.

2. If $t'_{(X,Y)} = t_X$, then $\text{Curry}([t']s) = [t](\text{Curry}(s))$.

3. $\text{apply}([t]r, [t]s) = [t](\text{apply}(r, s))$.

4. $\text{Apply}([t]s, G) = [t](\text{Apply}(s, G))$.

Proof: 1.

$$\begin{aligned} \text{curry}([t']s')_X(p)(b) &= ([t']s')_X(p, b) \\ &= s'_X(p, b, t'_X(p, b)) \\ &= s_X(p, t_X(p), b) \\ &= \text{curry}(s)_X(p, t_X(p))(b) \\ &= ([t](\text{curry}(s)))_X(p)(b). \end{aligned}$$

2.

$$\begin{aligned} \text{Curry}([t']s)_X(x)_Y &= s_{(X,Y)}(x, t'_{(X,Y)}(x)) \\ &= s_{(X,Y)}(x, t_X(x))_Y \\ &= [t](\text{Curry}(s))_X(x)_Y. \end{aligned}$$

3.

$$\begin{aligned} \text{apply}([t]r, [t]s)_X(x) &= ([t]r)_X(x)([t]s)_X(x) \\ &= (r_X(x, t_X(x)))(s_X(x, t_X(x))) \\ &= \text{apply}(r, s)_X(x, t_X(x)) \\ &= [t](\text{apply}(r, s))_X(x). \end{aligned}$$

4.

$$\begin{aligned} \text{Apply}([t]s, G)_X(x) &= ([t]s)_X(x)_{G(X)} \\ &= s_X(x, t_X(x))_{G(X)} \\ &= \text{Apply}(s, G)_X(x, t_X(x)) \\ &= [t](\text{Apply}(s, G))_X(x). \blacksquare \end{aligned}$$

Notation: Suppose

$$\begin{aligned} P, K : (\mathbf{D}^{\text{EP}})^m &\rightarrow \mathbf{D}^{\text{EP}} \text{ and} \\ F : (\mathbf{D}^{\text{EP}})^m \times \mathbf{D}^{\text{EP}} &\rightarrow \mathbf{D}^{\text{EP}} \end{aligned}$$

are continuous functors and

$$t \in \Pi((P \circ \text{Fst}) \Rightarrow F),$$

then we define a continuous section

$$[K]t \in \Pi(P \Rightarrow (F \circ \langle \text{Id}_{(\mathbf{D}^{\text{EP}})^m}, G \rangle))$$

by setting

$$([K]t)_X(x) = \text{Apply}(\text{Curry}(t), K)_X(x) = t_{(X, G(X))}(x). \quad \blacksquare$$

We will need the following Lemma later:

Lemma 15 1. $\text{curry}([K]t) = [K](\text{curry}(t))$.

2. If $t'_{(X, Z, Y)} = t_{(X, Y, Z)}$ for each X, Y and Z , then $\text{Curry}([K \circ \text{Fst}]t') = [K](\text{Curry}(t))$.

3. $\text{apply}([K]s, [K]t) = [K](\text{apply}(s, t))$.

4. $\text{Apply}([K]t, H \circ \langle \text{Id}, K \rangle) = [K](\text{Apply}(t, H))$.

Proof: 1.

$$\begin{aligned} \text{curry}([K]t)_X(x)(y) &= ([K]t)_X(s, y) \\ &= t_{(X, K(X))}(x, y) \\ &= \text{curry}(t)_{(X, K(X))}(x)(y) \\ &= [K](\text{curry}(t))_X(x)(y). \end{aligned}$$

2.

$$\begin{aligned} \text{Curry}([K \circ \text{Fst}]t)_X(x)_Z &= \text{Curry}([K \circ \text{Fst}]t')_{(X, Z)} \\ &= t'_{(X, Z, G(X))}(x) \\ &= t_{(X, G(X), Z)}(x) \\ &= \text{Curry}(t)_{(X, G(X))}(x)_Z \\ &= [K](\text{Curry}(t))_X(x)_Z. \end{aligned}$$

3.

$$\begin{aligned} \text{apply}([K]s, [K]t)_X(x) &= (([K]s)_X(x))([K]t)_X(x) \\ &= (s_{(X, K(X))}(x))(t_{(X, K(X))}(x)) \\ &= (\text{apply}(s, t))_{(X, K(X))}(x) \\ &= [K](\text{apply}(s, t))_X(x). \end{aligned}$$

4.

$$\begin{aligned} \text{Apply}([K]t, H \circ \langle \text{Id}, K \rangle) &= ([K]t)_X(x)_{H(X, K(X))} \\ &= t_{(X, K(X))}(x)_{H(X, K(X))} \\ &= \text{Apply}(t, H)_{(X, K(X))}(x) \\ &= [K](\text{Apply}(t, H))_X(x). \quad \blacksquare \end{aligned}$$

5 Syntax of the polymorphic λ -calculus.

The types of the polymorphic λ -calculus are given by the following abstract syntax:

$$\sigma ::= \sigma_1 \Rightarrow \sigma_2 \mid \alpha \mid \Pi\alpha. \sigma$$

and the terms of the calculus are described as follows:

$$M ::= x \mid \lambda x : \sigma. M \mid M_1(M_2) \mid \Lambda\alpha. M \mid M\{\sigma\}.$$

We distinguish a subcollection of *well-typed* terms of the calculus to be those terms M for which $\vdash M : \sigma$ is derivable from the typing rules listed below. The sequents in the typing rules are of the form $H \vdash_{\Sigma} M : \sigma$ where $H = x_1 : \sigma_1, \dots, \sigma_n$ is a (possibly empty) list of typings for variables which must include all of the free term variables of M , and $\Sigma = \alpha_1, \dots, \alpha_n$ is a list of type variables which must include all of the free type variables that appear in $\sigma_1, \dots, \sigma_n$ and M . We use $\vdash_{\Sigma} M$ as an abbreviation for $H \vdash_{\Sigma} M$ where H is the empty list and $H \vdash M$ as abbreviation for $H \vdash_{\Sigma} M$ where Σ is the empty list.

Typing rules for the polymorphic λ -calculus.

projection:

$$H_1, x : \sigma, H_2 \vdash_{\Sigma} x : \sigma$$

\Rightarrow introduction:

$$\frac{H, x : \sigma_1 \vdash_{\Sigma} M : \sigma_2}{H \vdash_{\Sigma} \lambda x : \sigma_1. M : \sigma_1 \Rightarrow \sigma_2}$$

Π introduction:

$$\frac{H \vdash_{\Sigma, \alpha} M : \sigma}{H \vdash_{\Sigma} \Lambda\alpha. M : \Pi\alpha. \sigma}$$

\Rightarrow elimination:

$$\frac{H \vdash_{\Sigma} M_1 : \sigma_1 \Rightarrow \sigma_2 \quad H \vdash_{\Sigma} M_2 : \sigma_1}{H \vdash_{\Sigma} M_1(M_2) : \sigma_2}$$

Π elimination:

$$\frac{H \vdash_{\Sigma} M : \Pi\alpha. \sigma_1}{H \vdash_{\Sigma} M\{\sigma_2\} : [\sigma_2/\alpha]\sigma_1}$$

Restrictions:

- In the projection rule, the variable x does not appear in H_1 or H_2 .
- In the Π introduction rule, there is no free occurrence of α in the type of any variable in H .
- In the Π elimination rule, all free variables of σ_2 are in Σ .

The terms of the calculus (in particular, the well-typed terms) are taken to satisfy a collection of equational rules of the form $H \vdash_{\Sigma} M_1 = M_2$ where H and Σ are lists of variable typings and type variables as described above. Again, we assume that H lists all of the free term variables that appear in M and Σ includes all of the free type variables that appear in H and M . The rules are given as follows:

Equational rules for the polymorphic λ -calculus.

reflexivity:

$$H_1, x : \sigma, H_2 \vdash_{\Sigma} x = x : \sigma$$

ξ :

$$\frac{H, x : \sigma_1 \vdash_{\Sigma} M_1 = M_2 : \sigma_2}{H \vdash_{\Sigma} \lambda x : \sigma_1. M_1 = \lambda x : \sigma. M_2 : \sigma_1 \Rightarrow \sigma_2}$$

type ξ :

$$\frac{H \vdash_{\Sigma, \alpha} M_1 = M_2 : \sigma}{H \vdash_{\Sigma} \Lambda \alpha. M_1 = \Lambda \alpha. M_2 : \Pi \alpha. \sigma}$$

congruence:

$$\frac{H \vdash_{\Sigma} M_1 = M_2 : \sigma_1 \quad H \vdash_{\Sigma} M_3 = M_4 : \sigma_1 \Rightarrow \sigma_2}{H \vdash_{\Sigma} M_3(M_1) = M_4(M_2) : \sigma_2}$$

type congruence:

$$\frac{H \vdash_{\Sigma} M_1 = M_2 : \Pi \alpha. \sigma_1}{H \vdash_{\Sigma} M_1\{\sigma_2\} = M_2\{\sigma_2\} : [\sigma_2/\alpha]\sigma_1}$$

It is not difficult to see that from these rules, a lambda expression M satisfies $H \vdash_{\Sigma} M : \sigma$ if and only if it satisfies $H \vdash_{\Sigma} M = M : \sigma$. Thus, for the remaining axioms, we use $H \vdash_{\Sigma} M : \sigma$ as an abbreviation for $H \vdash_{\Sigma} M = M : \sigma$.

symmetry:

$$\frac{H \vdash_{\Sigma} M_1 = M_2 : \sigma}{H \vdash_{\Sigma} M_2 = M_1 : \sigma}$$

transitivity:

$$\frac{H \vdash_{\Sigma} M_1 = M_2 : \sigma \quad H \vdash_{\Sigma} M_2 = M_3 : \sigma}{H \vdash_{\Sigma} M_1 = M_3 : \sigma}$$

β :

$$\frac{H, x : \sigma_1 \vdash_{\Sigma} M_2 : \sigma_2 \quad H \vdash_{\Sigma} M_1 : \sigma_1}{H \vdash_{\Sigma} (\lambda x : \sigma_1. M_2)(M_1) = [M_1/x]M_2 : \sigma_2}$$

type β :

$$\frac{H \vdash_{\Sigma, \alpha} M : \sigma_1}{H \vdash_{\Sigma} (\Lambda \alpha. M)\{\sigma_2\} = [\sigma_2/\alpha]M : [\sigma_2/\alpha]\sigma_1}$$

η :

$$\frac{H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2}{H \vdash_{\Sigma} \lambda x : \sigma_1. M(x) = M : \sigma_1 \Rightarrow \sigma_2}$$

type η :

$$\frac{H \vdash_{\Sigma} M : \Pi \alpha. \sigma}{H \vdash_{\Sigma} \Lambda \alpha. M\{\alpha\} = M : \Pi \alpha. \sigma}$$

Restrictions:

- In the reflexivity axiom, the variable x does not appear in H_1 or H_2 .
- In the type ξ rule, there is no free occurrence of α in the type of a variable in H .
- In the type β rule, there is no free occurrence of α in the type of a variable in H .
- In the η rule, the variable x does not occur free in M .
- In the type η rule, the variable α does not occur free in M .

6 Semantics of the polymorphic λ -calculus.

In this section we provide a detailed description of a semantics for the polymorphic λ -calculus, whose syntax was described in the previous section. We end by showing that our model interprets types differently from the models based on finitary projections described earlier and we show that the equational theory of our model is different from that of any such model.

If $m \geq i \geq 1$, then define $\mathbf{P}^{i,m} : (\mathbf{D}^{\text{EP}})^m \rightarrow \mathbf{D}^{\text{EP}}$ to be the i 'th projection, *i.e.* the continuous functor whose action on objects is given by $\mathbf{P}^{i,m}(D_1, \dots, D_m) = D_i$ and whose action on arrows is $\mathbf{P}^{i,m}(f_1, \dots, f_m) = f_i$.

If $\Sigma = \alpha_1, \dots, \alpha_m$ is a list of type variables then $\mathfrak{S}[\vdash_{\Sigma} \sigma]$ will be a continuous functor from $(\mathbf{D}^{\text{EP}})^m$ into \mathbf{D}^{EP} . The semantic function $\mathfrak{S}[_]$ is defined inductively as follows:

- $\mathfrak{S}[\vdash_{\alpha_1, \dots, \alpha_m} \alpha_i] = \mathbf{P}^{i,m}$
- $\mathfrak{S}[\vdash_{\Sigma} \sigma_1 \Rightarrow \sigma_2] = \mathfrak{S}[\vdash_{\Sigma} \sigma_1] \Rightarrow \mathfrak{S}[\vdash_{\Sigma} \sigma_2]$
- $\mathfrak{S}[\vdash_{\Sigma} \Pi \alpha. \sigma] = \Pi^m(\mathfrak{S}[\vdash_{\Sigma, \alpha} \sigma])$

We also assign a meaning to a sequent $H \vdash_{\Sigma} \sigma$ by the equation:

$$[\sigma_1, \dots, \sigma_n \vdash_{\Sigma} \sigma] = \#(\mathfrak{S}[\vdash_{\Sigma} \sigma_1], \dots, \mathfrak{S}[\vdash_{\Sigma} \sigma_n]) \Rightarrow \mathfrak{S}[\vdash_{\Sigma} \sigma]$$

Example: The type of the polymorphic identity is given as follows:

$$\begin{aligned} \mathfrak{S}[\vdash \Pi \alpha. \alpha \Rightarrow \alpha] &= \Pi^1(\mathfrak{S}[\vdash_{\alpha} \alpha \Rightarrow \alpha]) \\ &= \Pi^1(\mathfrak{S}[\vdash_{\alpha} \alpha] \Rightarrow \mathfrak{S}[\vdash_{\alpha} \alpha]) \\ &= \Pi^1(\mathbf{P}^{1,1} \Rightarrow \mathbf{P}^{1,1}) \quad \blacksquare \end{aligned}$$

We now define the semantics of the sequents of the calculus. In general, the value

$$[x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash_{\Sigma} M : \sigma]$$

will be a continuous section of the functor

$$[\sigma_1, \dots, \sigma_n \vdash_{\Sigma} \sigma] : (\mathbf{D}^{\text{EP}})^n \rightarrow \mathbf{D}^{\text{EP}}.$$

The semantic equations are given as follows:

- $\llbracket x_1 : \sigma_1, \dots, x_i : \sigma_i, \dots, x_n : \sigma_n \vdash_\Sigma x_i : \sigma_i \rrbracket = \mathbf{p}^{i,n}$
- $\llbracket H \vdash_\Sigma \lambda x : \sigma_1. M : \sigma_1 \Rightarrow \sigma_2 \rrbracket = \mathbf{curry}(\llbracket H, x : \sigma_1 \vdash_\Sigma M : \sigma_2 \rrbracket)$
- $\llbracket H \vdash_\Sigma \Lambda \alpha. M : \Pi \alpha. \sigma \rrbracket = \mathbf{Curry}(\llbracket H \vdash_\Sigma, \alpha M : \sigma \rrbracket)$
- $\llbracket H \vdash_\Sigma M_2(M_1) : \sigma_2 \rrbracket = \mathbf{apply}(\llbracket H \vdash_\Sigma M_2 : \sigma_1 \Rightarrow \sigma_2 \rrbracket, \llbracket H \vdash_\Sigma M_1 : \sigma_1 \rrbracket)$
- $\llbracket H \vdash_\Sigma M\{\sigma_2\} : [\sigma_2/\alpha]\sigma_1 \rrbracket = \mathbf{Apply}(\llbracket H \vdash_\Sigma M : \Pi \alpha. \sigma_1 \rrbracket, \llbracket \vdash_\Sigma \sigma_2 \rrbracket)$

For the second equations, one must suppose that the variable x doesn't appear free in H . To see that the third line makes sense, we note the following:

Lemma 16 *If α does not appear free in the type σ , then $\mathfrak{S}[\vdash_\Sigma, \alpha \sigma] = \mathfrak{S}[\vdash_\Sigma \sigma] \circ \mathbf{Fst}$.*

Proof: Straightforward structural induction on σ . ■

Example: The polymorphic identity function is the following continuous section of $\Pi^1(\mathbf{P}^{1,1} \Rightarrow \mathbf{P}^{1,1})$:

$$\begin{aligned} \llbracket \vdash \Lambda \alpha. \lambda x : \alpha. x : \Pi \alpha. \alpha \Rightarrow \alpha \rrbracket &= \mathbf{Curry}(\llbracket \vdash_\alpha \lambda x : \alpha. x : \alpha \Rightarrow \alpha \rrbracket) \\ &= \mathbf{Curry}(\mathbf{curry}(\llbracket x : \alpha \vdash_\alpha x : \alpha \rrbracket)) \\ &= \mathbf{Curry}(\mathbf{curry}(\mathbf{p}^{1,1})). \quad \blacksquare \end{aligned}$$

Lemma 17 (Permutation) *If we have*

$$\begin{aligned} \{1, \dots, n\} &= \{i_1, \dots, i_n\} \text{ and} \\ \{1, \dots, m\} &= \{j_1, \dots, j_m\} \end{aligned}$$

then

$$\begin{aligned} \llbracket x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash_{\alpha_1, \dots, \alpha_n} M : \sigma \rrbracket_{(X_1, \dots, X_m)}(p_1, \dots, p_n) \\ = \llbracket x_{i_1} : \sigma_{i_1}, \dots, x_{i_n} : \sigma_{i_n} \vdash_{\alpha_{j_1}, \dots, \alpha_{j_m}} M : \sigma \rrbracket_{(X_{j_1}, \dots, X_{j_m})}(p_{i_1}, \dots, p_{i_n}) \end{aligned}$$

Proof: Easy structural induction on M . ■

Lemma 18 (Substitution) *Suppose $H \vdash_\Sigma M_1 : \sigma_1$ and $H, x : \sigma_1 \vdash_\Sigma M_2 : \sigma_2$, then*

$$\mathbf{apply}(\mathbf{curry}(\llbracket H, x : \sigma_1 \vdash_\Sigma M_2 : \sigma_2 \rrbracket), \llbracket H \vdash_\Sigma M_1 : \sigma_1 \rrbracket) = \llbracket H \vdash_\Sigma [M_1/x]M_2 \rrbracket.$$

Proof: To help reduce the amount of notation needed for the arguments below, let

$$\begin{aligned} r &= \llbracket H \vdash_\Sigma [M_1/x]M_2 \rrbracket \\ s &= \llbracket H, x : \sigma_1 \vdash_\Sigma M_2 : \sigma_2 \rrbracket \\ t &= \llbracket H \vdash_\Sigma M_1 : \sigma_1 \rrbracket \end{aligned}$$

We must show that $r = [t]s$. Let n and m be the lengths of H and Σ respectively. The proof is by structural induction on the term M_2 . There are six cases.

Case 1: $M_2 \equiv y \neq x$. Suppose y is the i 'th variable in H . Then $r = \llbracket H \vdash_{\Sigma} y : \sigma_2 \rrbracket = \mathbf{p}^{i,n} = [t](\mathbf{p}^{i,n+1}) = t$.

Case 2: $M_2 \equiv x$. We have $r = t$ and $[t]s = [t](\mathbf{p}^{n+1,n+1}) = t$, so $r = [t]s$.

Case 3: $M_2 \equiv \lambda y : \sigma. M$. Suppose that $\sigma_2 \equiv \sigma \Rightarrow \tau$ so that $H, y : \sigma \vdash_{\Sigma} M : \tau$.

$$\begin{aligned}
r &= \llbracket H \vdash_{\Sigma} \lambda y : \sigma. [M_1/x]M : \sigma_2 \rrbracket \\
&= \text{curry}(\llbracket H, y : \sigma \vdash_{\Sigma} [M_1/x]M : \sigma \rrbracket) \\
&= \text{curry}(\llbracket [H, y : \sigma \vdash_{\Sigma} M_1 : \sigma_1] \rrbracket \llbracket H, y : \sigma, x : \sigma_1 \vdash_{\Sigma} M : \sigma \rrbracket) \quad (\text{hyp}) \\
&= [t](\text{curry}(\llbracket H, x : \sigma_1, y : \sigma \vdash_{\Sigma} M : \sigma \rrbracket)) \quad (\text{Lemmas 14.1 and 17}) \\
&= [t]s.
\end{aligned}$$

Case 4: $M_2 \equiv \Lambda \alpha. M$. Suppose that $\sigma_2 \equiv \Pi \alpha. \sigma$ so that $H \vdash_{\Sigma, \alpha} M : \sigma$.

$$\begin{aligned}
r &= \llbracket H \vdash_{\Sigma} \Lambda \alpha. [M_1/x]M : \sigma_2 \rrbracket \\
&= \text{Curry}(\llbracket H \vdash_{\Sigma, \alpha} [M_1/x]M : \sigma \rrbracket) \\
&= \text{Curry}(\llbracket [H \vdash_{\Sigma, \alpha} M_1 : \sigma_1] \rrbracket \llbracket H, x : \sigma_1 \vdash_{\Sigma, \alpha} M : \sigma \rrbracket) \quad (\text{hyp}) \\
&= [t](\text{Curry}(\llbracket H, x : \sigma_1 \vdash_{\Sigma, \alpha} M : \sigma \rrbracket)) \quad (\text{Lemmas 14.2 and 17}) \\
&= [t]s.
\end{aligned}$$

Case 5: $M_2 \equiv M(N)$. Suppose that $H \vdash_{\Sigma} M : \sigma \Rightarrow \sigma_2$ and $H \vdash_{\Sigma} N : \sigma$.

$$\begin{aligned}
r &= \llbracket H \vdash_{\Sigma} ([M_1/x]M)([M_1/x]N) : \sigma_2 \rrbracket \\
&= \text{apply}(\llbracket H \vdash_{\Sigma} [M_1/x]M : \sigma \Rightarrow \sigma_2 \rrbracket, \llbracket H \vdash_{\Sigma} [M_1/x]N : \sigma \rrbracket) \\
&= \text{apply}([t]\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \sigma \Rightarrow \sigma_2 \rrbracket, [t]\llbracket H, x : \sigma_1 \vdash_{\Sigma} N : \sigma \rrbracket) \quad (\text{hyp}) \\
&= [t](\text{apply}(\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \sigma \Rightarrow \sigma_2 \rrbracket, \llbracket H, x : \sigma_1 \vdash_{\Sigma} N : \sigma \rrbracket)) \\
&= [t]s.
\end{aligned}$$

Case 6: $M_2 \equiv M\{\sigma\}$. Suppose $H \vdash_{\Sigma} M : \tau$.

$$\begin{aligned}
r &= \llbracket H \vdash_{\Sigma} ([M_1/x]M)\{\sigma\} : \sigma_2 \rrbracket \\
&= \text{Apply}(\llbracket H \vdash_{\Sigma} [M_1/x]M : \tau \rrbracket, \llbracket \vdash_{\Sigma} \sigma \rrbracket) \\
&= \text{Apply}([t]\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \tau \rrbracket, \llbracket \vdash_{\Sigma} \sigma \rrbracket) \quad (\text{hyp}) \\
&= [t](\text{Apply}(\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \tau \rrbracket, \llbracket \vdash_{\Sigma} \sigma \rrbracket)) \quad (\text{Lemma 14.4}) \\
&= [t]s. \quad \blacksquare
\end{aligned}$$

Lemma 19 $\llbracket \vdash_{\Sigma} [\sigma_2/\alpha]\sigma_1 \rrbracket = \llbracket \vdash_{\Sigma} \sigma_1 \rrbracket \circ \langle \text{Id}, \llbracket \vdash_{\Sigma} \sigma_2 \rrbracket \rangle$.

Proof: Structural induction on σ_1 . \blacksquare

Lemma 20 (Type Substitution) Suppose $H \vdash_{\Sigma, \alpha} M : \sigma_1$, and α does not appear free in H , then

$$\text{Apply}(\text{Curry}(\llbracket H \vdash_{\Sigma, \alpha} M : \sigma_1 \rrbracket, \llbracket \vdash_{\Sigma} \sigma_2 \rrbracket)) = \llbracket H \vdash_{\Sigma} [\sigma_2/\alpha]M : [\sigma_2/\alpha]\sigma_1 \rrbracket.$$

Proof: To help reduce the amount of notation needed, let

$$\begin{aligned} s &= \llbracket H \vdash_{\Sigma} [\sigma_2/\alpha]M : [\sigma_2/\alpha]\sigma_1 \rrbracket \\ t &= \llbracket H \vdash_{\Sigma, \alpha} M : \sigma_1 \rrbracket \\ K &= \llbracket \vdash_{\Sigma} \sigma_2 \rrbracket. \end{aligned}$$

We must show that $s = [K]t$. The proof is by structural induction on M . There are five cases.

Case 1: $M \equiv x$. This is trivial.

Case 2: $M \equiv \lambda y : \sigma. N$. Suppose $\sigma_1 \equiv \sigma \Rightarrow \tau$ so that $N : \tau$.

$$\begin{aligned} s &= \llbracket H \vdash_{\Sigma} \lambda y : [\sigma_2/\alpha]\sigma. [\sigma_2/\alpha]N : [\sigma_2/\alpha]\sigma_1 \rrbracket \\ &= \text{curry}(\llbracket H, y : [\sigma_2/\alpha]\sigma \vdash_{\Sigma} [\sigma_2/\alpha]N : [\sigma_2/\alpha]\tau \rrbracket) \\ &= \text{curry}([K]\llbracket H, y : \sigma \vdash_{\Sigma, \alpha} N : \tau \rrbracket) && \text{(hyp)} \\ &= [K](\text{curry}(\llbracket H, y : \sigma \vdash_{\Sigma, \alpha} N : \tau \rrbracket)) && \text{(Lemma 14.1)} \\ &= [K]t. \end{aligned}$$

Case 3: $M \equiv \Lambda\beta. N$. Suppose that $\sigma_1 \equiv \Pi\beta. \sigma$ so that $N : \sigma$.

$$\begin{aligned} s &= \llbracket H \vdash_{\Sigma} \Lambda\beta. [\sigma_2/\alpha]N : [\sigma_2/\alpha]\sigma_1 \rrbracket \\ &= \text{Curry}(\llbracket H \vdash_{\Sigma, \beta} [\sigma_2/\alpha]N : [\sigma_2/\alpha]\sigma_1 \rrbracket) \\ &= \text{Curry}([K \circ \text{Fst}]\llbracket H \vdash_{\Sigma, \beta, \alpha} N : \sigma_1 \rrbracket) && \text{(hyp)} \\ &= [K](\text{Curry}(\llbracket H \vdash_{\Sigma, \beta} N : \sigma_1 \rrbracket)) && \text{(Lemmas 15.2 and 17)} \\ &= [K]t. \end{aligned}$$

Case 4: $M \equiv N_1(N_2)$. Suppose that $N_1 : \sigma \Rightarrow \sigma_1$ and $N_2 : \sigma$.

$$\begin{aligned} s &= \llbracket H \vdash_{\Sigma} ([\sigma_2/\alpha]N_1)([\sigma_2/\alpha]N_2) : [\sigma_2/\alpha]\sigma_1 \rrbracket \\ &= \text{apply}(\llbracket H \vdash_{\Sigma} [\sigma_2/\alpha]N_1 : [\sigma_2/\alpha](\sigma \Rightarrow \sigma_1) \rrbracket, \llbracket H \vdash_{\Sigma} [\sigma_2/\alpha]N_2 : [\sigma_2/\alpha]\sigma \rrbracket) \\ &= \text{apply}([K]\llbracket H \vdash_{\Sigma, \alpha} N_1 : (\sigma \Rightarrow \sigma_1) \rrbracket, [K]\llbracket H \vdash_{\Sigma, \alpha} N_2 : \sigma \rrbracket) && \text{(hyp)} \\ &= [K](\text{apply}(\llbracket H \vdash_{\Sigma, \alpha} N_1 : (\sigma \Rightarrow \sigma_1) \rrbracket, \llbracket H \vdash_{\Sigma, \alpha} N_2 : \sigma \rrbracket)) && \text{(Lemma 15.3)} \\ &= [K]t. \end{aligned}$$

Case 5: $N\{\sigma\}$. Suppose $H \vdash_{\Sigma} N : \tau$.

$$\begin{aligned} s &= \llbracket H \vdash_{\Sigma} ([\sigma_2/\alpha]N)\{[\sigma_2/\alpha]\sigma\} : [\sigma_2/\alpha]\sigma_1 \rrbracket \\ &= \text{Apply}(\llbracket H \vdash_{\Sigma} [\sigma_2/\alpha]N : [\sigma_2/\alpha]\tau \rrbracket, \llbracket \vdash_{\Sigma} [\sigma_2/\alpha]\sigma \rrbracket) \\ &= \text{Apply}([K]\llbracket H \vdash_{\Sigma, \alpha} N : \tau \rrbracket, \llbracket \vdash_{\Sigma} [\sigma_2/\alpha]\sigma \rrbracket) && \text{(hyp)} \\ &= \text{Apply}([K]\llbracket H \vdash_{\Sigma, \alpha} N : \tau \rrbracket, \llbracket \vdash_{\Sigma} \sigma \rrbracket \circ \langle \text{Id}, K \rangle) && \text{(Lemma 19)} \\ &= [K](\text{Apply}([K]\llbracket H \vdash_{\Sigma, \alpha} N : \tau \rrbracket, \llbracket \vdash_{\Sigma} \sigma \rrbracket)) && \text{(Lemma 15.4)} \\ &= [K]t. \quad \blacksquare \end{aligned}$$

Lemma 21 Suppose $H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2$. If x does not appear in H , then

$$\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2 \rrbracket = \llbracket H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2 \rrbracket \circ \text{fst}.$$

Proof: By structural induction on M . ■

The following is a more dramatic version of Lemma 16:

Lemma 22 Suppose $H \vdash_{\Sigma} M : \sigma$. If $\alpha \notin \Sigma$, then $\llbracket H \vdash_{\Sigma, \alpha} M : \sigma \rrbracket = \llbracket H \vdash_{\Sigma} M : \Pi\alpha. \sigma \rrbracket \circ \text{Fst}$.

Proof: By structural induction on M . ■

We will say that an equation $H \vdash_{\Sigma} M_1 = M_2 : \sigma$ is satisfied by our semantics just in case $\llbracket H \vdash_{\Sigma} M_1 : \sigma \rrbracket = \llbracket H \vdash_{\Sigma} M_2 : \sigma \rrbracket$. We are now prepared to state our central result:

Theorem 23 The semantic function $\llbracket _ \rrbracket$ satisfies the rules for the polymorphic λ -calculus.

Proof: There are eleven rules altogether. Those whose proofs are non-trivial are the rules β , type β , η and type η . The β rule and type β rule are immediate from the Substitution Lemma (18) and Type Substitution Lemma (20) respectively.

First we consider the η rule:

$$\frac{H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2}{H \vdash_{\Sigma} \lambda x : \sigma_1. M(x) = M : \sigma_1 \Rightarrow \sigma_2}$$

This is subject to the restriction that the variable x does not occur free in M (and hence does not appear in H). We have

$$\begin{aligned} & \llbracket H \vdash_{\Sigma} \lambda x : \sigma_1. M(x) : \sigma_1 \Rightarrow \sigma_2 \rrbracket \\ &= \text{curry}(\llbracket H, x : \sigma_1 \vdash_{\Sigma} M(x) : \sigma_2 \rrbracket) \\ &= \text{curry}(\text{apply}(\llbracket H, x : \sigma_1 \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2 \rrbracket, \text{snd})) \\ &= \text{curry}(\text{apply}(\llbracket H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2 \rrbracket \circ \text{fst}, \text{snd})) && \text{(Lemma 21)} \\ &= \llbracket H \vdash_{\Sigma} M : \sigma_1 \Rightarrow \sigma_2 \rrbracket \end{aligned}$$

We now prove the type η rule:

$$\frac{H \vdash_{\Sigma} M : \Pi\alpha. \sigma}{H \vdash_{\Sigma} \Lambda\alpha. M\{\alpha\} = M : \Pi\alpha. \sigma}$$

This is subject to the restriction that the variable α does not occur free in M (and hence does not appear in Σ).

$$\begin{aligned} & \llbracket H \vdash_{\Sigma} \Lambda\alpha. M\{\alpha\} : \Pi\alpha. \sigma \rrbracket \\ &= \text{Curry}(\llbracket H \vdash_{\Sigma, \alpha} M\{\alpha\} : \sigma \rrbracket) \\ &= \text{Curry}(\text{Apply}(\llbracket H \vdash_{\Sigma, \alpha} M : \Pi\alpha. \sigma \rrbracket, \llbracket \vdash_{\Sigma, \alpha} \alpha \rrbracket)) \\ &= \text{Curry}(\text{Apply}(\llbracket H \vdash_{\Sigma} M : \Pi\alpha. \sigma \rrbracket \circ \text{Fst}, \text{Snd})) && \text{(Lemma 22)} \\ &= \llbracket H \vdash_{\Sigma} M : \Pi\alpha. \sigma \rrbracket. \quad \blacksquare \end{aligned}$$

Example: We wish to compute the interpretation $\mathfrak{I}[\Pi\alpha. \alpha]$ of the trivial type. This will show that our model is distinct from the finitary projection model (and also that the equational theories are distinct, since the equation $\lambda(x : \Pi\alpha. \alpha). \lambda(y : \Pi\alpha. \alpha). x = \lambda(x : \Pi\alpha. \alpha). \lambda(y : \Pi\alpha. \alpha). y$ is valid in our model and not in the finitary projection model).

Let $\langle t_X \rangle$ be a continuous section of the identity functor. For all $f \in \mathbf{D}^{\text{EP}}(X, Y)$, we get $f^L(t_X) \leq t_Y$. Given an arbitrary domain X , let us consider $Y = X + X$ (the coalesced sum), with the two morphisms (that are left adjoints) $\text{inl} : X \rightarrow Y$ and $\text{inr} : X \rightarrow Y$. Let fl (resp. fr) be the morphism in \mathbf{D}^{EP} corresponding to inl (resp. inr). Then, we must have $F(fl)^L(t_X) \leq t_Y$ and $F(fr)^L(t_X) \leq t_Y$ which entails $t_Y = \perp$, and then $t_X = \perp$. ■

7 A model of Type:Type.

There are two purposes of this section. Firstly, we want to illustrate the notion of a family of domains indexed over a domain with the example of domains over a universal domain. Secondly, we want to explain how the finitary projection model of [1] relates to our model. In order to illustrate the first point, we shall actually show that the finitary projection model is a model for a more powerful type system than second-order type system, namely a type system with a type of all types. A more categorical description of this model may be found in [29].

7.1 A reformulation of Type:Type

The system we use is an extension of intuitionistic type theory [15]¹, where we add one universe, but with a slight change in the axioms for type equalities as compared with the version in [15].

We suppose that we have a special type U , which should be thought of as a type of indices for types, and an operation T over the element of U , to be regarded as a dependent type over U . We suppose that there exists an element u of type U such that $T(u) = U$, that is, a name for the type of all types.

We suppose furthermore that there is an “internalisation” of the product operation of dependent types. Namely, there exists

- $\pi : \Pi_{a:U}.(T(a) \rightarrow U) \rightarrow U$,
- $\text{App} : \Pi_{a:U}. \Pi_{b:T(a) \rightarrow U}. T(\pi(a, b)) \rightarrow (\Pi_{x:T(a)}. T(b(x)))$,
- $\text{Lambda} : \Pi_{a:U}. \Pi_{b:T(a) \rightarrow U}. (\Pi_{x:T(a)}. T(b(x))) \rightarrow T(\pi(a, b))$.

We ask that these operations are inverses, that is $\text{Lambda} \circ \text{App} = \text{id}$, and $\text{App} \circ \text{Lambda} = \text{id}$.² The ordinary formulation [15] is with a type equality rule $T(\pi(a, b)) = \Pi_{x:T(a)}. T(b(x))$, but this rule

¹Notice that it should be possible, from the interpretation of the dependent product and sums over a domain outlined in the previous section, to give an interpretation of intuitionistic type theory in terms of Scott domains (see [16]). We shall not develop this here, since the precise verification that it is indeed a model is similar to checking that we get a model for second-order type theory, and we have given this verification in full detail.

²It is interesting to note that this system is that obtained by representing the $\text{Type} : \text{Type}$ calculus in the LF-framework [10], and also that it may be seen as providing a syntactic condition for what it means to be a model of $\text{Type} : \text{Type}$ following the ideas of [3].

does not seem to square with a “standard” semantics. For our purpose, the “weaker” system with only isomorphisms is sufficient. It is significant that the $Type : Type$ system, even with this weaker form of equality, can be translated syntactically into our formalism (in particular, it is possible to interpret Girard’s paradox [7] in it, and so all types are “syntactically” inhabited).

Rather than describe this syntactic translation in full formal details, let us give some examples. The universal type of second-order λ -calculus $\Pi\alpha.\alpha \rightarrow \alpha$ is first translated by $\Pi\alpha : Type. \Pi x : \alpha. \alpha$ in the $Type : Type$ system. Then, it becomes $T(\pi(u, \lambda x. \pi(x, \lambda y. x)))$. And so, if M is of this type, and N is of type $T(u)$ (that is N is a type), we can form the application of M to N by $App(u, \lambda x. \pi(x, \lambda y. x), M, N)$. In the same way, the type $\Pi\alpha. \alpha$ will be interpreted by $T(\pi(u, id))$. Since App and $Lambda$ are inverses, the β - η -conversion rules will be satisfied.

7.2 Semantics in domain theory

We can point at once to one important difference between the finitary projection model and our categorical model. In it, types are not interpreted directly as arbitrary domains, but as finitary projections of a single “universal domain”. So, for the construction of this model, we must first pick a domain D so that $[D \rightarrow D]$ is embedded in D by the pair (Φ, Ψ) (as is well-known following Scott, such domains can, for instance, be built using an inverse limit construction). It is important to note that there are many such domains, that there is nothing canonical in this choice, and that the influence of this choice over the model is not clear. This is, however, the only part that is “non canonical” in the construction.

Let D be a domain so that there exists an embedding-projection pair (Φ, Ψ) of $[D \rightarrow D]$ into D . An element $p \in D \rightarrow D$ is called a finitary projection if, and only if, $p \leq id$, $p \circ p = p$, and the image of p is a domain with respect to the restriction of the order on D . It is known that the partial order of finitary projections (with respect to the extensional ordering) is a domain, that we shall write \mathbf{Fp} , and that this domain is embedded in $[D \rightarrow D]$ [22]. We obtain an embedding-projection pair (Φ_0, Ψ_0) from \mathbf{Fp} into D , from the composition of this embedding-projection from \mathbf{Fp} into $[D \rightarrow D]$ with (Φ, Ψ) . We now take for the interpretation of the set U the image of Φ_0 , which we again call U . This should cause no real confusion. Notice that we do not interpret the type of types U by the “universal” domain D .

In the sequel, it will be convenient to use the “uncurried” notation “ $f(x, y)$ ” for “ $f(x)(y)$ ”. If $a \in U$, then a defines a finitary projection $\Psi_0(a)$ and hence a subdomain of D , namely the image of this finitary projection $T(a) = \{x \in D \mid \Psi_0(a)(x) = x\}$. Notice that $T(a)$ is a subdomain of the “universal domain” D . Furthermore, $a \in U$, and that if $a \leq b$ in U then $T(a)$ is a subdomain of $T(b)$. The family $T(x)$, $x \in U$, is a good example of a continuous family of domains over a domain.

Each $T(a)$, for $a \in U$, is embedded in the “universal domain” D , where the embedding is the inclusion map, and the projection is defined by $x \mapsto \Psi_0(a, x)$. If $b \in T(a) \rightarrow U$, since $D \rightarrow D$ is embedded into D , there is a “canonical” embedding of $\Pi_{x:T(a)}.T(b(x))$ into D . Explicitly, the embedding is defined in the following way: let $f \in \Pi_{x:T(a)}.T(b(x))$, then the image of f under this embedding is defined by $x \mapsto f(\Psi_0(x, a))$. The definition of the projection is: for $f \in D \rightarrow D$, the image of f under the projection is defined by $x \mapsto \Psi_0(b, f(x))$. This embedding will define an element of \mathbf{Fp} , hence an element of U by Φ_0 , that we shall write as $\pi(a, b)$. Explicitly, we have $\pi(a, b) =$

$\Phi(\lambda x. \Phi(\lambda z. \Psi(b(\Psi(a, z)), \Psi(x, \Psi(a, z)))))$. By construction, we have that $T(\pi(a, b))$ is isomorphic to $\Pi_{x:T(a)}. T(b(x))$ and App , $Lambda$ are notation for the two halves of this isomorphism. We find that, if $c \in T(\pi(a, b))$, and $d \in T(a)$, then $App(c, d) = \Psi(c, d)$, and if $c \in \Pi_{x:T(a)}. T(b(x))$, then $Lambda(c) = \Phi(\lambda x. c(\Psi_0(a, x)))$.

We can then check the desired equalities. For $c \in T(\pi(a, b))$ we have $c = \Phi(\Psi(c) \circ (\Psi_0(a)))$. Indeed, we have

$$\begin{aligned} c &= \Psi(\pi(a, b), c) \\ &= \Phi(\lambda z. \Psi(b(\Psi(a, z)), \Psi(c, \Psi(a, z)))). \end{aligned}$$

Hence $\Psi(c) = \lambda z. \Psi(b(\Psi(a, z)), \Psi(c, \Psi(a, z)))$ and $\Psi(c) \circ \Psi_0(a) = \Psi(c)$ since $\Psi(a) \circ \Psi_0(a) = \Psi(a)$, because $a \in U$, so that

$$\begin{aligned} Lambda(App(c)) &= \Phi(\Psi(c) \circ \Psi_0(a)) \\ &= \Phi(\Psi(c)) \\ &= c. \end{aligned}$$

For the other equality, we suppose that $c \in \Pi_{x:T(a)}. T(b(x))$, and then

$$\begin{aligned} App(Lambda(c)) &= \Psi(\Phi(c \circ (\Psi_0(a)))) \\ &= c \circ \Psi_0(a) \\ &= c. \end{aligned}$$

Finally, we build an element $u \in U$ so that $T(u) = U$. We take $u = \Phi_0(\Phi_0 \circ \Psi_0)$. Since $\Phi_0 \circ \Psi_0 \in \mathbf{Fp}$, we have $u \in U$. And $x \in T(u)$ if, and only if, $x \in D$ and $\Phi_0(\Psi_0(x)) = x$, hence if, and only if, $x \in U$. By definition of equality of domain, we get $T(u) = U$.

Since one can interpret second-order λ -calculus in this calculus, we get a model for second-order λ -calculus (and the reader can check that what we get in this way is indeed the model described in [1]).

7.3 An example

As an example, we shall show that, in general, the interpretation of $\Pi\alpha.\alpha$, which here is $T(\pi(u, id))$, is a non-trivial domain. This is significant because it shows that we get an essentially different model with the categorical approach, since there the interpretation of $\Pi\alpha.\alpha$ is the trivial domain. Since $T(\pi(u, id))$ is isomorphic to $\Pi_{x:U}. T(x)$, it is enough to show that $\Pi_{x:U}. T(x)$ is not trivial if U is not trivial (that is if D is not trivial). Let $a \in U$ be an element different from \perp . Then, if $x \in U$, we have $\Psi(x, a) \in T(x)$, by definition of $T(x)$. It results that $\lambda x. \Psi(x, a) \in \Pi_{x:U}. T(x)$, and we have $\lambda x. \Psi(x, a) \neq \perp$ since $a \neq \perp$.

The intuitive explanation of the difference between the models is that in the finitary projection model we restrict ourselves to domains that are finitary projections of a given “big” domain, and the only morphisms we allow are inclusions (and not arbitrary embeddings). We thus get a small category that is isomorphic to the domain $\mathbf{Fp}(D)$ of finitary projections over D . This category is a subcategory (but not a full one) of the category $\mathbf{D}^{\mathbf{EP}}$ via the inclusion functor. A dependent type becomes a continuous function f from $\mathbf{Fp}(D) = U$ into itself which defines, by composition with this inclusion functor, a dependent

domain over the domain U . We can then see that the general definition of the product of a dependent domain given previously will specialise itself to $T(\pi(u, f))$. This explains why the interpretation of $\Pi\alpha.\alpha$ is bigger in the finitary projection model: when we consider $\mathbf{Fp}(D)$ as “the” category of domains, we forget the morphisms that are not inclusions (for instance, non-trivial automorphisms). In a sense, the categorical model is a refinement of this model where we take into account embeddings that are not inclusions.

8 Questions and comparisons with related work.

We want first to describe why Girard’s model [8], [4] follows the same pattern as our present model. The idea is to translate all our definitions to the stable framework of [2]. That is, instead of requiring the continuity of functors and functions, we require further that pull-backs are preserved, a property called *stability*. In place of the extensional ordering on functions, we take the stable ordering. In place of natural transformations between functors we take cartesian natural transformations. We can then work in the category \mathbf{DI}^{EP} [2,8], or in the full subcategories of qualitative domains or coherent spaces [8]. The relationship with the work of J.Y. Girard is then explained by a general result due to E. Moggi, which we state in the following special case:

Proposition 24 *Let F be a stable functor from \mathbf{DI}^{EP} to \mathbf{DI}^{EP} , then a family $\langle t_X \rangle_{X \in \mathbf{DI}^{\text{EP}}}$ is a continuous and stable section of F if, and only if, it is uniform, that is $F(f)^R(t_Y) = t_X$ whenever $f \in \mathbf{DI}^{\text{EP}}(X, Y)$.*

We need first to express what a stable section is. A simple calculation of pull-backs in the Grothendieck fibration of F shows that (f, g, u, v) is a pull-back diagram, with $f \in (T, t) \rightarrow (X, x)$, $g \in (T, t) \rightarrow (Y, y)$, $u \in (X, x) \rightarrow (Z, z)$ and $v \in (Y, y) \rightarrow (Z, z)$ (that is, $f \in \mathbf{DI}^{\text{EP}}(T, X)$, $g \in \mathbf{DI}^{\text{EP}}(T, Y)$, $u \in \mathbf{DI}^{\text{EP}}(X, Z)$, $v \in \mathbf{DI}^{\text{EP}}(Y, Z)$, and $F(f)^L(t_T) \leq t_X$, $F(g)^L(t_T) \leq t_Y$, $F(u)^L(t_X) \leq t_Z$ and $F(v)^L(t_Y) \leq t_Z$), if, and only if, $t_T = F(f)^R(t_X) \wedge F(g)^R(t_Y)$. The key fact is that if $f \in \mathbf{DI}^{\text{EP}}(X, Y)$ then we can always find a domain Z and two morphisms $u, v \in \mathbf{DI}^{\text{EP}}(Y, Z)$ such that they form a pull-back diagram. This is clear if we think in terms of the representation using event structures of dI-domains (see section 3 of [4]). By expressing the stability condition for this diagram, we get the uniformity of $\langle t_X \rangle$.

The stable model leads to a “smaller” interpretation. For instance, in all the known stable models, the interpretation of $\Pi\alpha.\alpha \rightarrow \alpha$ is the two-point domain. In the model presented in this paper, this turns out to be infinite since it contains the following “continuous” operations indexed by an integer n : $f_X(x) = x$ if x bounds more than n finite elements, and $f_X(x) = \perp$ if x does not bound more than n finite elements (these are examples of “parametric” operations that are not uniform). It is not clear whether or not these “non uniform” operations are interesting. It seems that all the terms we get from the syntax of the second-order λ -calculus are uniform, and so the stable model may be helpful in producing fully abstract models.

A question raised by the last example is whether or not the interpretation of a given syntactic type is an effectively given domain [26]. We do not even know actually what is the precise form of the interpretation of $\Pi\alpha.\alpha \rightarrow \alpha$ (are there other elements than the ones given?). This question may be asked of the stable models too [8,4]. It was one of the motivations in introducing the notion of coherent domain

[8], since, in this case it is possible to give an “explicit” description of the interpretation of the syntactic types.

An important general question is the connection between these “models” and the general definition of a model for second-order λ -calculus given in [3]. A surprising point is that, strictly speaking, the present model, and Girard’s models as well, are not models in the sense of Bruce and Meyer (this was pointed out to us by E. Moggi). Indeed, it seems essential that the collection of types is interpreted as a *category*, and not as a *set*. This cannot be done if we follow verbatim the Bruce and Meyer definition. This is to be contrasted with the finitary projection model of [1], which is a model for Bruce and Meyer definition. This adds weight to the proposal of Seely of a more general definition of model [24,5], and, indeed, our construction is a model [5] in his sense. It would be also possible to generalise slightly the definition of Bruce and Meyer following the ideas developed in [2], so that this definition becomes equivalent to Seely’s definition.

We may ask also what are the relationship with other known models for polymorphisms. For instance, the ideal model of [13], or models in the effective topos (see for instance [11]). In contrast with the effective topos model [11], our model is a direct extension of that commonly used in denotational semantics of programming languages and it allows us to handle recursion at all types.

In our construction, we made the choice to use the category of embedding-projection pairs rather than arbitrary left adjoints. The constructions go through in the same way for with this category in place of embeddings. For instance, we get a simple model by taking complete algebraic lattices and left adjoints, model where the interpretation of the polymorphic identity type has only three points, as expected (see [5] for a brief description of this model). We do not understand the relationship between this model and the one presented in detail here. Notice that this choice does not appear in the stable case (as noticed by A. Pitts), due to the following remark: if a stable function $f : D \rightarrow D$ is greater than id_D for the stable ordering, then, this function is equal to the identity. Indeed, we have, for $x \in D$, $x \leq f(x)$ hence, by stability, $x = f(x) \wedge id_D(f(x))$, that is, $x = f(x)$. From this, we deduce that a left adjoint is, in the stable case, an embedding.

We have explained the central role Grothendieck fibrations and continuous sections play in the interpretation of polymorphism. Our presentation has been deliberately based on examples, and on one model in particular; a new model for polymorphism has been worked out in considerable detail. From another point of view, we have probably not been abstract enough. It is not yet clear what the right framework is in which to encompass and relate the full range of models, and what techniques to use to home-in on the model appropriate to meet certain requirements like full-abstraction.

References

- [1] Amadio, R., Bruce, K. B., Longo, G., *The finitary projection model for second order lambda calculus and solutions to higher order domain equations*. In: **Logic in Computer Science**, edited by A. Meyer, IEEE Computer Society Press, 1986, pp. 122–130.
- [2] Berry, G., *Stable models of typed λ -calculi*. In: **Fifth International Colloquium on Automata, Languages and Programs**, Springer-Verlag, **Lecture Notes in Computer Science**, vol. 62, 1978, pp. 72–89.
- [3] Bruce, K. and Meyer, A., *The semantics of polymorphic lambda-calculus*. In: **Semantics of Data Types**, edited by G. Kahn, D.B. MacQueen and G. Plotkin, **Lecture Notes in Computer Science**, vol. 173, Springer-Verlag, 1984, pp. 131–144.
- [4] Coquand Th., Gunter C. and Winskel G., *dl-domains as a model of polymorphism*. To appear in the proceedings of the Third Workshop on the Mathematical Foundations of Programming Language Semantics, New Orleans, LA 1987.
- [5] Coquand, T., and Ehrhard, T., *An equational presentation of higher-order logic*. In the proceedings of the conference “Category theory and computer science”, Edinburgh, September 1987, Springer Lecture Notes in Computer Science.
- [6] Fairbairn, J., *Design and implementation of a simple typed language based on the lambda-calculus*. University of Cambridge Computer Laboratory Technical Report, no. 75, 1985, 107pp.
- [7] Girard, J. Y., *Interprétation fonctionnelle et élimination des coupures de l’arithmétique d’ordre supérieur*. Thèse d’Etat, Université Paris VII, 1972.
- [8] Girard, J. Y., *The system F of variable types, fifteen years later*. **Theoretical Computer Science**, vol. 45, 1986.
- [9] Gunter, C. A., *Universal profinite domains*. **Information and Computing**, vol. 72 (1987), pp. 1–30.
- [10] R. Harper, F. Honsell, G. Plotkin. *A Framework for Defining Logics*. LICS, 1987.
- [11] Hyland, J.M.E. *The Effective Topos*. in The L.E.J. Brouwer Centenary Symposium, North-Holland, 1982.
- [12] Winskel, G. and Larsen, K., *Using information systems to solve recursive domain equations effectively*. In the proceedings of the conference on Abstract Datatypes, Sophia-Antipolis, France in June 1984, Springer Lecture Notes in Computer Science, 173.
- [13] Macqueen, D., Sethi, R., Plotkin, G.D., *An Ideal Model For Recursive Polymorphic Types*. POPL, 1984.
- [14] Martin-Löf, P., *An Intuitionistic Theory of Types*. Unpublished manuscript, 1971.

- [15] Martin-Löf, P., *Intuitionistic Type Theory*. Bibliopolis, 1980.
- [16] Martin-Löf, P., **The domain interpretation of type theory**. Lecture notes from the workshop on semantics of programming languages, Göteborg, 1983.
- [17] McCracken, N., **An Investigation of a Programming Language with a Polymorphic Type Structure**, Doctoral Dissertation, Syracuse University, 1979.
- [18] Pitts, A., *Polymorphism is set-theoretic constructively*. In the proceedings of the conference “Category theory and computer science”, Edinburgh, September 1987, Springer Lecture Notes in Computer Science.
- [19] Plotkin, G.D., *Complete partial orders, a tool for making meanings*. Lecture notes for the Pisa Summerschool, 1978.
- [20] Reynolds, J. C., *Polymorphism is not set-theoretic*. In: **Semantics of Data Types**, edited by G. Kahn, D.B. MacQueen and G. Plotkin, **Lecture Notes in Computer Science**, vol. 173, Springer-Verlag, 1984, pp. 145–156.
- [21] Reynolds, J. C., *Towards a theory of type structures*. In: **Colloque sur la Programmation**, Springer-Verlag, **Lecture Notes in Computer Science** 19, 1974, pp. 408–425.
- [22] Scott, D. S., *Some ordered sets in computer science*. In: **Ordered Sets**, edited by I. Rival., D. Reidel Publishing Company, 1981, pp. 677–718.
- [23] Scott, D. S., *Domains for Denotational Semantics*. ICALP 1982, Springer Lecture notes in Computer Science, 140, 1982.
- [24] Seely, R., *Categorical semantics for higher order polymorphic lambda calculus*. Manuscript, 1986, 33pp.
- [25] Smith, J., *Non derivability of Peano Axioms in Type Theory without Universes*. Unpublished manuscript.
- [26] Smyth, M., *Effectively given domains*. Theoretical Computer Science, vol. 5, pp. 257–274, 1977.
- [27] Smyth, M., *The largest cartesian-closed category of domains*. Theoretical Computer Science, 1983.
- [28] Smyth, M. B. and Plotkin, G. D., *The category-theoretic solution of recursive domain equations*. **SIAM Journal of Computing**, vol. 11 (1982), pp. 761–783.
- [29] Taylor, P., **Recursive domains, indexed category theory and polymorphism**. Ph.D. Thesis in mathematics, University of Cambridge, 1987.