



HAL
open science

Building continuous webbed models for System F

Stefano Berardi, Chantal Berline

► **To cite this version:**

Stefano Berardi, Chantal Berline. Building continuous webbed models for System F. Theoretical Computer Science, 2004, 315, pp.3-34. hal-00003750

HAL Id: hal-00003750

<https://hal.science/hal-00003750>

Submitted on 3 Jan 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Building continuous webbed models for System F

S. Berardi and C. Berline

October 15th 1998, revised November 15th 2000.

Abstract

We present here a large family of concrete models for Girard and Reynolds polymorphism (System F), in a non categorical setting. The family generalizes the construction of the model of Barbanera and Berardi [2], hence it contains complete models for $F\eta$ [5] and we conjecture that it contains models which are complete for F . It also contains simpler models, the simplest of them, \mathcal{E}^2 , being a second order variant of the Engeler-Plotkin model \mathcal{E} . All the models here belong to the continuous semantics and have underlying prime algebraic domains, all have the maximum number of polymorphic maps. The class contains models which can be viewed as two intertwined compatible webbed models of untyped λ -calculus (in the sense of [8]), but it is much larger than this. Finally many of its models might be read as two intertwined strict intersection type systems.

Contents

1	Introduction.	2
2	Polymax models of F.	8
2.1	Definition of polymax models.	8
2.2	Extra properties of our models.	10
2.3	Interpretation of System F in polymax models.	11
3	Preliminaries on prime webs and prime algebraic domains.	13
3.1	Prime webs.	13
3.2	Morphisms.	14
3.3	Subwebs.	15
3.4	Exponent.	15
3.5	Retraction pairs for application and abstraction.	15
3.6	Reflexive webs and webbed models of untyped λ -calculus.	16

4	Webbed models for F.	17
4.1	Definition of <i>Types</i> , <i>types</i> , Q and <i>AP</i> .	17
4.2	Definition of <i>Terms</i> .	18
4.3	Definition of \Rightarrow .	19
4.4	Definition of <i>apl</i> and <i>lbd</i> .	19
4.5	Definition of <i>Appl</i> and <i>Lambda</i> .	19
4.6	Conclusion.	21
5	Examples.	22
5.1	The square models.	22
5.1.1	The graph ² -models.	22
5.1.2	The \mathcal{E}^2 -model.	23
5.1.3	The (ext-K) ² -models.	24
5.2	The <i>BB</i> -model.	24
5.3	A candidate for β -completeness.	26
5.4	A candidate for being an intermediate model.	27
6	There is no stable analogue.	28
7	Miscellaneous.	29
7.1	Testing the conditions for completeness.	29
7.2	Comparison with the <i>u.r.</i> models.	30
7.3	Are all the conditions defining F -webs necessary ?	32

1 Introduction.

Girard's System F [16], which is the fragment of λ -calculus which can be controlled by second order propositional logic, dates back to [14],[15]. It was rediscovered independently by Reynolds [29] in a Computer Science setting, where it is also called polymorphism. In this paper we assume that the reader is familiar with the syntax of F , as presented in [16].

We present here a rather general family of concrete models for System F , which are based on prime algebraic domains and continuous functions, differ from previous models, and for which the interpretation of second order quantification is transparent and requires no functorial notion. In fact working with a particular model requires no category theory at all.

This concrete family generalizes the construction of the model of Barbanera and Berardi [2], called here the *BB*-model for short, which was shown to be complete for $F\eta$ in [5] and was indeed the first nonsyntactic complete model exhibited for this system. It also contains simpler models. The simplest model, called \mathcal{E}^2 here, is based on Engeler-Plotkin's model \mathcal{E} [12],[28], which will be called here "Engeler's model" for short.

We then compare the present class with the models proposed previously for F , at least with those models for which, from our point of view, comparison

makes sense. This comparison supposes some familiarity with those previously known models.

Which kind of models are the models presented here ? As already mentioned, the models here belong to the continuous semantics and they are built within the category of prime algebraic domains.

Within each model there are domains *Types* and *Terms*, where *F*-types and *F*-terms are interpreted, and second-order quantification ranges of course over *Types*. Furthermore the domain $\forall X.F(X)$, which interprets quantification relative to a morphism $F:Types \rightarrow Types$, contains all morphisms $f:Types \rightarrow Terms$ such that $f(X) \in F(X)$. Such models will be called *polymax* in the sequel, to remind us that they do not constrain polymorphic maps more than strictly necessary for modelling *F* (a useful formal definition of “polymax” is proposed in section 2.1, which however does not pretend to cover all intuitively polymax models). By *polymorphic map* we understand any element of a domain $\forall X.F(X)$, which is in turn called a (*semantic*) *polymorphic type*.

Restricting polymorphic types is often considered as desirable for studying terms as programs (see for example O’Hearn’s survey [26] on parametricity), but for a general study of polymorphism one has to be more permissive. Having no restriction over polymorphic maps was for example crucial for proving the completeness of the *BB*-model, and hence the existence of a nonsyntactic complete model for *F*.

Note that although all our models can be seen as very rich in polymorphic maps, they are not equivalent in this respect : for example the *BB*-model has “case functions” recognizing whether a semantic type is an arrow or a \forall , but this is not true of all our models (cf. section 7.1).

Our family contains models which, like \mathcal{E}^2 , can be viewed as two intertwined compatible webbed models of untyped lambda-calculus (the same model taken twice in the case of \mathcal{E}^2), but this is not the general case : a counter-example is the *BB*-model.

Let us now explain what we understand by “*webbed models*”. This terminology was introduced in [8] for referring to the models of untyped λ -calculus which 1. were based on a prime algebraic domain \mathcal{D} and, 2. could be described by a pair (W, j) , where W was the structured set of the prime elements of \mathcal{D} (its “*prime web*”) and j was a map making W a reflexive object in an adequate category of prime webs, the nature of the category depending on the semantics we are working in. For the continuous semantics, and for our concern, a prime web is a triple (D, \preceq, \approx) where D is a nonempty set, \preceq is a preorder and \approx is a reflexive and symmetric binary relation, both relations being compatible in a very natural sense (cf. Section 3). Working in the same spirit we get here a notion of *webbed models of F*. Such a model is describable from an *F*-web, which is obtained, roughly speaking, by intertwining two compatible prime webs, one of which is reflexive.

Similarly as for models of untyped λ -calculus, *the relations can all be taken as trivial* and we get then the simplest examples of webbed models of *F*, \mathcal{E}^2

being the simplest of all.

In examples we also isolate the subclass of *square models of F* . Such a model arises when the reflexive web can play the two roles, and the simplest example is once more \mathcal{E}^2 . Starting from a webbed model of untyped λ -calculus it is not always the case that it will give rise to a square model, but it is often the case (see Section 5.1).

Roughly speaking prime webs are to (binary) prime algebraic Scott domains what Scott’s information systems [33] are to Scott domains in the continuous semantics (cf. also [20]), and prime event structures to DI-domains in the stable semantics (see for ex. [10]) but, categorically speaking, the situation is less neat here (remark 3 in section 3.2).

Interest of the present class. The first interest of the class lies in the simplicity of the interpretations of terms in its models, and the second lies in the great variety of models that can be built, within a single *c.c.c.* It is for example easy to model in our setting the extensions of F considered in [10] (they include various constructors like product, sum and fixed points); products can even be already found in the simplest model \mathcal{E}^2 . It is also worth mentioning that all possible ¹ recursive equations on types admit solutions in all our models.

In particular, this class could be a good place to test the possible consistency of extensions of F , as well as the independence of added axioms and rules ². However we can’t really decide this point now since (in contrast to the webbed models of untyped λ -calculus) it is not clear to us at the moment of writing whether our webbed models for F can be significantly different from the point of view of equations between *pure F -terms*.

Concerning the classical *higher-order extensions of F* we mention that all our models can also be viewed as models of F_ω and that square models with trivial coherence (e.g. \mathcal{E}^2) can be viewed as models of the calculus of constructions plus “*kinds = types*”, but this is another story which is developed in [6].

Concerning completeness. In the present framework the construction of the *BB*-model appears natural rather than ad-hoc. Moreover our family is the most natural place to test the feasibility and necessity of the sufficient conditions for completeness w.r.t. $F\eta$ that we gave in [5], and to hierarchize them w.r.t. easiness (see sections 7.1). Finally it is a natural place to look for complete models for F , and we suggest a candidate ³.

These models might also prove useful for studying the syntax of F . This assertion relies on observations made by many authors in the context of webbed models for untyped λ -calculus ⁴.

¹where “possible” only refers to the obvious limitation that the type constructors occurring in the equation must exist in the model.

²For a survey of results of this kind obtained with webbed models of untyped λ -calculus see [8].

³Proving completeness for non extensional models will however require a new idea w.r.t. the proof in [5], since logical relations cannot distinguish between elements with the same applicative behaviour, and, hence, between two η -equivalent terms.

⁴See [8] for a survey, and for a direct use of webbed models in such matters.

Finally, an open question raised in [30] is whether a domain theoretic model of parametric polymorphism exists. A variant of this question (cf. [26]) is whether a cpo model of polymorphic λ -calculus can be modified to be parametric. Once more the present class could prove useful, since the simplicity of some of its models should allow easier manipulations.

Polymax models. To describe the way in which our models indeed model F , we could have used the transparent abstract definition of models that we introduced in [5]. We prefer to use here a variant, which is a little more general and gives an explicit role to a constant *types* which is relevant for practical constructions. Both variants cover only polymax models. The one proposed here covers all known concrete polymax models, namely : all *universal retraction (u.r.) models*, whether continuous or stable, all *trivial* models of F (which interpret F -terms in a model of untyped λ -calculus, by forgetting all typing considerations), and finally all the continuous models that we build in this paper. The definition of [5] excludes those *u.r.* models which are *not* based on projections or closures. The difference between these definitions is the following (in a domain setting): in [5] we were working with a domain *Types* whose elements were subdomains of *Terms*, while here we only require that *Types* is a family of subdomains of *Terms* indexed by a domain *types*.⁵

If the indexing is 1-1, which is the case with our models, then this definition is equivalent to the first one and furthermore *types* can then be eliminated, which leaves us with the exact definition of [5]. When the indexing *is not* 1-1 syntactic types must be interpreted in *types*. When the indexing *is* 1-1 *Types* can be given a structure isomorphic to *types* and the interpretation of types can then also be done in *Types*.

Examples of *non* (intuitively) polymax models are : PER and realizability models, Girard's stable model and its CGW continuous or stable variants; these models will be briefly discussed in the sequel.

Axiom C. A convenient, but rather coarse, indicator of the richness in polymorphic maps is the Axiom (Scheme) C isolated by Longo-Milsted-Soloviev in [21]. Axiom (Scheme) C is the set of equations $t^{\forall\alpha.\sigma}\tau = t^{\forall\alpha.\sigma}\tau'$, for all F -types τ, τ', σ such that α is not free in σ , and for all F -terms $t^{\forall\alpha.\sigma}$. Thus C asserts that constant polymorphic types only contain constant polymorphic maps.

Obviously a nontrivial polymax model can't satisfy C (were trivial only means here that *Types* is a singleton), in particular our continuous models will not satisfy C . In fact none of the models mentioned below that belong to the continuous semantics will satisfy C , with the exception of *PER* and realizability models (when built on a partial combinatory algebra which belongs

⁵For webbed models *types* is a domain whose elements are webs and the elements of *Types* are the domains generated by these webs. In the case of universal retraction models *types* is a domain of retractions, of some fixed domain M , while the elements of *Types* are the subdomains of M whose underlying sets are the ranges of these retractions. Thus using *types* will allow to refer, when useful, to a lower level of the resulting polymax model.

to the continuous semantics). These last models have constant polymorphic maps.

Comparison with Girard’s and CGW-models. Girard’s categorical model [16] belongs to the stable semantics and quantifies over qualitative domains. The continuous CGW-model [11] is its adaptation to the continuous semantics, and there is finally a stable CGW model, which follows the same lines and replaces qualitative domains by DI-domains [10]. The three models are extensional (models of $F\eta$). None is polymax and the two stable models satisfy C . These models also encode morphisms (and functors) via traces, but they are much more constrained than our models, even in the stable case. The advantages of Girard’s and CGW’s models are mainly conceptual.

The first one is that in their setting second order quantification ranges over all (qualitative, DI- or Scott-) domains. But the price to pay for this universality is that there are heavy categorical conceptual tools to digest before just being able to interpret a type or a term, and we doubt that it is really possible to work with such models. One can also note that, as a consequence, one has at most one model in each c.c.c., in sharp contrast to our setting.

In fact the work in Girard (and CGW) shows that quantifying over a domain of finite domains (instead of the whole class of domains) would be sufficient. However such reduction would have no interest in their settings since functorial constraints would remain necessary. As a matter of fact functors are also used in Girard (in connection with stability) to restrict very strongly the interpretation of polymorphic types, and a similar but weaker restriction occurs in the continuous CGW.

Having a smaller interpretation of (some basic) polymorphic types is indeed the second advantage of these models; one can note however that none of them is parametric.

Girard’s and CGW’s models are trivially incomplete for $F\eta$, since there are types σ (such as $\forall\alpha.\alpha$) which they interpret as singletons (hence the model satisfies $x^\sigma = y^\sigma$). Another argument, which only works in the stable case, is that they satisfy Axiom C . By contrast many of our models are complete for $F\eta$ (cf. Section 7.1).

There is no similar class in the stable semantics. We developed our models in the continuous semantics. It was very natural to hope for an analogous class in the stable semantics. Unfortunately, and surprisingly, this is not possible, and we will see why in Section 6. It is easy to see, indeed, that our method, when worked out in the stable semantics, forces axiom C (which is rather orthogonal to being polymax, but is certainly pleasant). But, then, it is not very difficult to show that having a genuine *model* of F would lead us to adopt a much more stringent condition than axiom C , somewhat similar to Girard’s one, and the resulting model(s) would probably be as complex as Girard’s one. In other words, our models can be seen as a drastic simplification of the continuous CGW model, but there is no such simplification in the stable

case.

Comparing webbed models with u.r. models. Another interesting connection is the one with *universal retraction models* or *u.r. models*, which were introduced, for the continuous semantics, by D. Scott and McCracken in [31], [32], [23], [24] and continued by Amadio-Bruce-Longo [1]. In these models terms are interpreted as elements of a model of the untyped λ -calculus, and types are interpreted by retractions ranging over a suitable class. The word “universal” refers to the fact that in these models there is a type of all types. Then Berardi [4] showed that similar work could be done for the stable semantics, taking this time the whole class of stable retractions (see [7] for a survey).

We already mentioned that our continuous models are polymax, that this could not be true for their stable analogues (if any), and that in fact they had no stable analogues. In contrast *u.r.* models (exist and) are polymax in both the continuous and stable semantics.

It is possible to look for a deeper connection and to compare *u.r.* models with the *square-models* of Section 5.1, namely those webbed models arising from some (webbed) model M of untyped λ -calculus which is used “twice”. In this case we indeed also interpret both terms and types as elements of M . Superficially the similarity stops here, since we make no use of any universal retraction. We however deepen the comparison in Section 7.2 where we show that the interpretation of types and terms is simpler, and definitely different, in our setting.

Webbed models vs realizability and PER-models . It is worth to say a word about realizability and *PER* models, since these models are successfully used to study some programming aspects of F , and since Girard’s first model of F was Troelstra’s realizability model HRO_2 ([14],[15],[35]). However these models are rather orthogonal to ours since, as already mentioned, they realize few polymorphic maps while ours are polymax⁶.

A connection with intersection type systems. We end with a brief connection with intersection type systems. Engeler’s model is transparently equivalent to a strict intersection type system (called System E in [8], “ E ” for Engeler), which is a simplified version of Dezani et al.’s intersection type system [9]. Like all intersection type systems, System E is based on an intuitionistic implication and a “set theoretical” conjunction (in fact a finite conjunction is exactly a finite set in system E) ; *strict* means here that there is a restriction on the use of \rightarrow : only conjunctions which are singletons are allowed on the right handside of the arrow.

More generally many⁷ of the webbed models of untyped λ -calculus can be

⁶In particular the logic defined from any of our models is trivial (from a realizability point of view).

⁷“All models”, if we do not insist that the system be recursively presented.

seen (modulo a straightforward translation) as extensions of system E ⁸. Similarly, the models of system F which are presented here can be seen as an intertwined pair of compatible extensions of system E (each with its own notion of implication, etc.).

Plan of the paper. In Section 2 we give our formal definition of polymax models, and sketch the proof that they are indeed models of F . Section 3 gives the preliminaries on prime webs which are needed to understand why most of the conditions for defining webs for F in Section 4 are indeed unavoidable. In Section 4 we define these webs and show how they generate models. In Section 5 we give many examples and also raise some questions and conjectures about their equational theories. In Section 6 we show that our construction has no analogue in the stable semantics. In Section 7 we discuss three independent side points : comparison with *u.r.* models, $F\eta$ -completeness, and the role of some of the practical conditions on F -webs.

2 Polymax models of F .

Preliminaries. We recall that a *retraction pair* between two objects A, B of a category is, by definition, a pair (f, g) of morphisms such that $g \circ f = id_A$. We then say that A is a *retract* of B and that f is *left invertible*.

Given a category $Univ$ whose objects are sets (with additional structure) and morphisms are (specific) functions we will say that the nonempty object X is a *substructure* of the object A if $X \subseteq A$ and, for all pair (C, f) where C is an object and f a function from C to X , we have that f is a morphism iff $i \circ f$ is a morphism, where i is the inclusion map (this implies that i is a morphism). For example if $Univ$ is a category of Scott domains and continuous functions the second condition expresses that the order of X is the restriction of that of A , and that for all nonempty directed subset X' of X , the sup of X' in X remains the sup of X' in A . In the stable semantics one should add a similar condition for suitable inf's, and so on.

2.1 Definition of polymax models.

By a *polymax model of system F* we mean a tuple :

$$\mathcal{M} = \langle Univ, \langle types, Types, Terms \rangle, \langle \Rightarrow, lbd, apl \rangle, \langle Q, Lambda, Appl \rangle \rangle$$

such that :

1. $Univ$ is a cartesian closed category with enough points [3]. Thus, we may think that the objects of $Univ$ are real sets (plus additional structure), and

⁸by addition of sets of (possibly recursive) equations between types, and a possible notion of subtyping, moreover the use of conjunction can be restricted to sets of compatible types.

morphisms are real functions. We assume furthermore that the objects of $Univ$ are nonempty, as *sets*. The reader can think of $Univ$ as a *c.c.c* of Scott domains and continuous functions.

By definition of cartesian closed, $Univ$ is equipped with a cartesian product \times , and contains, for each pair of objects (A, B) , an object $A \rightarrow B$ representing $Hom(A, B)$ and an evaluation morphism $eval_{A,B}$. For sake of readability we assume here that $Hom(A, B)$ is the underlying set of $A \rightarrow B$, and that $eval$ is the usual application of a function to its argument.

2. $Terms$ is an object of $Univ$.

3. $types$ is an object of $Univ$.

The elements of $types$ will be denoted D, D' ⁹.

4. $Types$ is a family of substructures¹⁰ of $Terms$, indexed by $types$.

Its elements will be denoted X_D , or simply X .

5. $\Rightarrow \in Hom(types \times types, types)$.

The interpretation of syntactic types will be in $types$, and \Rightarrow will interpret the arrow constructor on syntactic types¹¹.

6. lbd and apl are indexed by $types \times types$ and, for each pair D, D' ,

$(lbd_{D,D'}, apl_{D,D'})$ is a retraction pair which makes $X_D \rightarrow X_{D'}$ a retract of $X_{D \Rightarrow D'}$.

The family (lbd, apl) will interpret abstraction over a term variable, and application of a term to a term.

7. $Q \in Hom(types \rightarrow types, types)$.

Second order quantification will be interpreted by Q , and $Q(F)$ will also be denoted $\forall D.F(D)$ for $F \in types \rightarrow types$.

8. $(Lambda, Appl)$ is a retraction pair which makes $types \rightarrow Terms$ a retract of $Terms$.

$Lambda$ and $Appl$ will interpret abstraction of terms over type-variables and application of terms to types.

⁹To fit previous papers or book. This is maybe a little misleading since the D 's are not domains.

¹⁰The authors are grateful to G. Rosolini for noticing that a previous version of point 4 was oversimplified.

¹¹As mentioned in the introduction, in case the indexing is 1-1, this interpretation can be transferred to $Types$ (cf. Section 2.3). Note that even in this case the distinction between " \rightarrow " and " \Rightarrow " will have to be done: in all the concrete examples $X \rightarrow Y$ and $X \Rightarrow Y$ can only be, in the best cases (e.g. extensionality), isomorphic.

9. For all $F \in Hom(types, types)$ we let now $Hom_F(types, Terms)$ be the subset of $Hom(types, Terms)$ containing those morphisms f such that $f(D) \in X_{F(D)}$ for all $D \in types$.

Our last requirement is that $Lambda$ sends elements of $Hom_F(types, Terms)$ into $X_{Q(F)}$, and conversely for $Appl$. The corresponding restriction of the pair $(Lambda, Appl)$ will be denoted $(Lambda_F, Appl_F)$ ¹².

Some justifications of the basic technical choices can be found in [5].

Out of any polymax model \mathcal{M} for F , and for any assignment ρ of elements of $types$ to type-variables of F , and of elements of $Terms$ to term variables of F , in a compatible way, we may define an interpretation $[\cdot]_\rho$ of F -types and F -terms. This is detailed in Section 2.3.

A polymax model \mathcal{M} of F identifies $(\alpha)\beta$ -convertible terms and α -convertible types. It identifies η -convertible terms, and is then labelled *extensional*, if we ask that (lbd, apl) is a family of inverse isomorphisms, and that $(Lambda_F, Appl_F)$ is a pair of inverse bijections. We also say in this case that \mathcal{M} is a *model of F_η* .

It is interesting to make the following observation:

Remark 1 (*Models of F_ω*) Suppose that Q is left-invertible, then the above interpretation of F extends canonically to an interpretation of F_ω [6].

2.2 Extra properties of our models.

In addition to the above properties, all the models we build in Section 4 satisfy:

1. $Univ$ is the *c.c.c.* of Scott domains (or prime algebraic Scott domains) and continuous functions.
2. $Terms = \cup Types$ (It is equivalent to be a term or to be an element of a type) (Lemma 8).
3. $Terms \subseteq types$ (Lemma 8).
4. The indexing $D \mapsto X_D$ is 1-1, so there is an order on $Types$ which makes $Types$ a domain isomorphic to $types$.
5. Q is left invertible, which implies in particular that, all our models, are models of F_ω .
6. There exists a polymorphic “trace function” $j : \forall X. \forall Y. X \Rightarrow Y$ such that $j(X, X, x) = x$ for all $X, x \in X$ (cf. Section 7.1).

Furthermore, many models satisfy:

7. \Rightarrow is (quasi) left invertible or/and
8. Q is an isomorphism or/and
9. $Terms = types$ and $Q = Lambda$.

Properties 5,6,7 are interesting in connection with completeness questions (cf. Section 7.1). Property 8 might be interesting in connection with the exploration of the possible equational theories of these models, and Property 9 is

¹²If $Univ$ is some standard ccc of domains, then conditions 8-9 imply that $Hom_F(Types, Terms)$ is a retract of $X_{Q(F)}$, via $(Lambda_F, Appl_F)$. But in general $Hom_F(Types, Terms)$ need not be an object of $Univ$.

related to the observation that the simplest of our models (including \mathcal{E}^2) also happen to be canonically models of the calculus of construction plus “*kinds = types*” [6] (cf. Section 5.1).

Warning. The construction of our models involves two categories, each with its own notion of morphisms and substructures, namely the category of Scott domains and, at a lower level, that of prime webs, that we will introduce later on. We insist on the fact that when we view these models as polymax models we mean that the ambient category is the upper level one, namely that *Univ is a category of domains*. One benefit of this choice, together with our definition of models from *types*, is that it allows clean comparisons with other models, especially with *u.r.* models (whose webs, if any, are not relevant).

2.3 Interpretation of System F in polymax models.

We assume here familiarity with the syntax of F , as presented in [16].

In the following *Vartypes* and *Varterms* denote respectively the set of type- and term- variables, and \mathcal{M} is a model in the sense of the previous section.

Interpretation of F -types. As already mentioned, syntactic types will be interpreted in *types*. When the correspondence between *types* and *Types* is 1-1, as it is the case with most practical models, including the family of this paper, this interpretation can be transferred to *Types*. The interest of this manipulation, which is handled at the end of this section, is that now syntactic types are directly interpreted as objects of the *c.c.c.* (domains for example), that *Types* can itself be viewed as an object of the *c.c.c.*, and that we can forget *types* (cf. also [5]).

Definition 1 A type-environment is a (total) function ρ from *Vartypes* to *types*.

Given a type-environment ρ and $D \in \text{types}$, we denote by $\rho[\alpha : D]$ the environment ρ' defined by : $\rho'(\alpha) := D$, and $\rho'(\alpha) = \rho(\alpha)$ otherwise.

Then $|\sigma|_\rho$ is defined, for all ρ , by induction on σ :

- $|\alpha|_\rho := \rho(\alpha)$
- $|\sigma \rightarrow \tau|_\rho := |\sigma|_\rho \Rightarrow |\tau|_\rho$
- $|\forall\alpha.\sigma|_\rho := Q(D \mapsto |\sigma|_{\rho[\alpha:D]})$.

Interpretation of F -terms.

Definition 2 An environment (for types and terms) is a partial function ρ from *Vartypes* \cup *Varterms* to *types* \cup *Terms* such that the restriction of ρ to *Vartypes* is a type-environment and for all term-variables $x^\sigma \in \text{dom}(\rho)$ we have $\rho(x^\sigma) \in X_{|\sigma|_\rho}$ (hence $\rho(x^\sigma) \in \text{Terms}$).

Given an environment ρ , a term-variable x^σ and a semantic term $v \in X_{|\sigma|_\rho}$, we denote by $\rho[x^\sigma : v]$ the environment ρ' which coincides with ρ everywhere but in x^σ , where $\rho'(x^\sigma) := v$. Given an environment ρ and an element D of *types* we define $\rho[\alpha : D]$ as the environment ρ' such that: $\text{dom}(\rho') = \text{dom}(\rho) - \{x^\tau / \alpha \text{ is free in } \tau\}$, $\rho'(\alpha) = D$, and $\rho'(\xi) = \rho(\xi)$ for all type or term-variable $\xi \in \text{dom}(\rho')$ which is different from α .

The interpretation of F -terms under all possible environments is by induction on the complexity of the term t and goes as follows :

- $|x^\sigma|_\rho := \rho(x^\sigma)$
- $|t^{\sigma \rightarrow \tau} u^\sigma|_\rho := \text{apl}_{|\sigma|_\rho, |\tau|_\rho}(|t^{\sigma \rightarrow \tau}|_\rho)(|u^\sigma|_\rho)$
- $|\lambda x^\sigma. t^\tau|_\rho := \text{abd}_{|\sigma|_\rho, |\tau|_\rho}(v \in X_{|\sigma|_\rho} \mapsto |t^\tau|_{\rho[x^\sigma : v]})$.
- $|t^{\forall \alpha. \sigma} \tau|_\rho := \text{Appl}(|t^{\forall \alpha. \sigma}|_\rho)(|\tau|_\rho)$
- $|\lambda \alpha. t^\sigma|_\rho := \text{Lambda}(D \in \text{types} \mapsto |t^\sigma|_{\rho[\alpha : D]})$.

The fact that this definition is correct mainly uses that *Univ* is a *c.c.c.*, that each X_D is a substructure of *Terms* and that our models are polymax. Let us be a little more precise: to show the correctness of the definition one has, as usual, to show simultaneously, by induction on the structure of t , that $|t^\sigma|_\rho$ is defined as soon as ρ is defined on the free type or term-variables of t and that it only depends on the values of ρ on these variables, and that $|t^\sigma|_\rho \in X_{|\sigma|_\rho}$ (hence, here, $|t^\sigma|_\rho \in \text{Terms}$). For this we need here the following lemma and corollary.

Lemma 3 *For all t^σ , for all ρ and all $\bar{\alpha}, \bar{x} := x_1^{\sigma_1}, \dots, x_n^{\sigma_n}$ such that the following makes sense,*

$$(\bar{D}, \bar{v}) \mapsto |t^\sigma|_{\rho[\bar{\alpha} : \bar{D}; \bar{x} : \bar{v}]} \text{ belongs to } \text{Hom}(\text{types}^{l(\bar{\alpha})} \times |X_{\sigma_1}|_\rho \times \dots \times |X_{\sigma_n}|_\rho, \text{Terms}).$$

The proof is routine, using the fact that we are in a *c.c.c.* and that we took enough care with our domains and codomains of morphisms. *However it is worth noting* that this is exactly the point where one uses the fact that each semantic type is a substructure of *Terms*. Note also that when dealing with the case of an application of a term to a term *it is crucial that we did not constrain polymorphic maps*.

Corollary 4 *For all t^σ , for all ρ, α, x^σ such that the following makes sense:*

1. $v \in X_{|\sigma|_\rho} \mapsto |t^\tau|_{\rho[x^\sigma : v]}$ is in $\text{Hom}(X_{|\sigma|_\rho}, X_{|\tau|_\rho})$
2. $D \in \text{types} \mapsto |t^\sigma|_{\rho[\alpha : D]}$ is in $\text{Hom}(\text{types}, \text{Terms})$.

Proof. Both are immediate, but the first one uses the fact that $X_{|\tau|_\rho}$ is a substructure of *Terms*.

Then we are able to prove :

Lemma 5 $\sigma =_{\alpha} \tau$ implies $\mathcal{M} \models |\sigma|_{\rho} = |\tau|_{\rho}$ for all environment ρ .
 $t =_{(\alpha)\beta} u$ implies $\mathcal{M} \models |t|_{\rho} = |u|_{\rho}$ for all environment ρ .
If \mathcal{M} is extensional, then the last sentence is also true for $\beta\eta$ -equivalent terms.

Proof. The proof is once again straightforward; the case of immediate reduction is done using the fact that all pairs $(lbd_{D,D'}, appl_{D,D'})$, and $(\text{Lambda}, \text{Appl})$ are retraction pairs (plus the adequate variation for the extensional case).

Interpreting syntactic types in $Ttypes$. We suppose now that \mathcal{T} , defined by $\mathcal{T}(D) = X_D$, is an injective map (from $types$ to $Ttypes$). Then we can use \mathcal{T} to transfer the structure of $types$ to $Ttypes$, which becomes then an object of $Univ$, while \mathcal{T} becomes an isomorphism. In case of domains it is of course enough to transfer the inclusion order of $types$ to a partial order on $Ttypes$. We now define :

$$X \Rightarrow Y := \mathcal{T}(\mathcal{T}^{-1}(X) \Rightarrow \mathcal{T}^{-1}(Y)), \text{ so that } X_D \Rightarrow X_{D'} := X_{D \Rightarrow D'}, \text{ and}$$

$$\forall X.F(X) := \forall D.(\mathcal{T}^{-1} \circ F \circ \mathcal{T})(D)$$

It is then clear how to modify the preceding interpretation (of types and terms) in such a way that the types be interpreted by elements of $Ttypes$ instead of $types$. In this case we get at the end : $|t^{\sigma}|_{\rho} \in |\sigma|_{\rho}$ instead of $|t^{\sigma}|_{\rho} \in X_{|\sigma|_{\rho}}$.

3 Preliminaries on prime webs and prime algebraic domains.

3.1 Prime webs.

Definition 6 A prime web is a triple $W := (D, \Leftrightarrow, \preceq)$ where D is a nonempty set, \Leftrightarrow is a reflexive and symmetric binary relation on D , and \preceq is a preorder on D , both relations being compatible in the sense that $x \Leftrightarrow y$ and $x' \preceq x$ and $y' \preceq y$ imply $x' \Leftrightarrow y'$.

Note that $x \preceq y$ implies $x \Leftrightarrow y$.

We consider respectively the relations \Leftrightarrow and \preceq as *trivial* if \Leftrightarrow is $\Omega \times \Omega$ and \preceq is equality. When dealing with examples in Section 5 trivial relations will not be written down explicitly.

A subset of D whose elements are pairwise related by \Leftrightarrow is called a *clique* (for \Leftrightarrow). We will denote by \downarrow the downward closure operator w.r.t. \preceq : $\downarrow a := \{x / \exists y \in a (x \preceq y)\}$. An equivalent formulation of the compatibility condition between \preceq and \Leftrightarrow is that the downward closure of a clique is a clique.

The equivalence relation induced by \preceq on D will be denoted \sim , hence $x \sim y$ iff $x \preceq y$ and $y \preceq x$.

We extend canonically the relations \preceq and \Leftrightarrow to subsets of D by : $a \preceq b$ iff $\downarrow a \subseteq \downarrow b$, and $a \Leftrightarrow b$ iff $a \cup b$ is a clique for \Leftrightarrow . Finally we let D_f^{\Leftrightarrow} be the set of finite cliques of D for \Leftrightarrow .

From a prime web $W := (D, \Leftrightarrow, \preceq)$ we can define a prime algebraic Scott domain, namely the ordered set $\mathcal{S}(W) := (S^{\downarrow \Leftrightarrow}(D), \subseteq)$, where $S^{\downarrow \Leftrightarrow}(D)$ is the set of closed cliques, namely the cliques a of W such that $a = \downarrow a$.

Remark 2 (not used in the sequel). $\mathcal{S}(W)$ is binary in the sense that any set B of prime elements of the domain is (upper) bounded as soon as its elements are pairwise bounded. Conversely, from any binary prime algebraic domain (\mathcal{D}, \leq) we recover a prime web $\mathcal{W}(\mathcal{D})$ just by taking for D the set of prime elements of \mathcal{D} , for \preceq the restriction of \leq to D , and by defining $x \approx y$ if $\{x, y\}$ is bounded in \mathcal{D} . The correspondence is such that $\mathcal{S}(\mathcal{W}(\mathcal{D}))$ is isomorphic to \mathcal{D} and $\mathcal{W}(\mathcal{S}(W))$ to W / \sim .

3.2 Morphisms.

Definition 7 A morphism from $W := (D, \approx, \preceq)$ to $W' := (D', \approx', \preceq')$ is any map j from D to D' which satisfies :

1. j is injective,
2. $x \approx y \iff j(x) \approx' j(y)$,
3. $j(x) \preceq' j(y) \implies x \preceq y$.

An embedding is a morphism j such that equivalence holds in 3.

An isomorphism is an embedding such that $\forall y \in D' \exists x \in D (y \sim' j(x))$.

Note that in the definition above “morphism” has to be taken in a categorical sense and not in the algebraic sense (where morphisms have, and only have, to preserve relations). On the other hand the definition of isomorphism above is what we need here, and is a little more general than being an invertible morphism (since j need not be surjective).

Notation. For any function j we let :

$$j^\bullet(u) := \{j(x) / x \in u\} \text{ and } j^{\bullet-1}(u) := \{x / j(x) \in u\}.$$

If j is a morphism from W to W' , then $\mathcal{S}(W)$ is a retract of $\mathcal{S}(W')$, since the maps i and p defined respectively on $S^{\downarrow \approx}(D)$ and $S^{\downarrow \approx'}(D')$ by $i(u) := \downarrow' (j^\bullet(u))$ and $p(v) = \downarrow j^{\bullet-1}(v)$, are obviously continuous and such that $p \circ i = id$. We let $\mathcal{S}(j) := \downarrow' \circ j^\bullet$ and $\mathcal{P}(j) := \downarrow \circ j^{\bullet-1}$. These maps are indeed more than continuous: there are *additive* in the sense that they commute with all existing unions, and not only directed ones (since this is already the case with $\downarrow, \downarrow', j^\bullet$ and $j^{\bullet-1}$).

If j is an embedding then $(\mathcal{S}(j), \mathcal{P}(j))$ is an embedding-projection pair between domains (i.e. a pair (i, p) such that $p \circ i = id$ and $i \circ p \leq id$).

Finally j is an isomorphism iff $\mathcal{S}(W)$ and $\mathcal{S}(W')$ are isomorphic under $(\mathcal{S}(j), \mathcal{P}(j))$.

Remark 3 (Not used in the sequel). Thus “morphisms” and “embeddings” give rise to two categories of prime webs. Since prime webs generate Scott domains it is natural to relate these categories to those concerning Scott domains. First we note that, since $\mathcal{S}(j)$ is indeed more constrained than just being additive, \mathcal{S} cannot be viewed as (half of) an equivalence of categories between (prime webs, morphisms) and (binary prime algebraic Scott domains, additive retraction pairs). However $(\mathcal{S}, \mathcal{W})$ is, essentially¹³, an equivalence of categories between (prime webs, embeddings) and (binary prime algebraic Scott domains,

¹³that is “up to \sim ”.

additive embedding-projection pairs). A last remark : prime webs are similar to event structures ¹⁴, at least as defined in [25], however the notions of morphisms that we use for generating continuous models of λ -calculus ¹⁵ is more general than the one proposed in the literature on event structures.

3.3 Subwebs.

We say that W is a *subweb* of W' if $D \subseteq D'$ and the relations on D are the restrictions to D of the relations on D' . If we already know that W and W' are both subwebs of some W'' , then W is a subweb of W' iff $D \subseteq D'$. Note also that W is a subweb of W' iff $D \subseteq D'$ and the inclusion is an embedding ; the corresponding embedding-projection pair (i, p) between $S(W)$ and $S(W')$ will be called *canonical* and we have $p(v) = v \cap D$.

3.4 Exponent.

From two prime webs $W := (D, \approx, \preceq)$ and $W' := (D', \approx', \preceq')$, we can define a prime web

$$\begin{aligned} W \Rightarrow W' &:= (D_f^{\hat{\approx}} \times D', \approx'', \preceq''), \text{ where} \\ (a, x) \preceq'' (b, y) &\iff b \preceq a \text{ and } x \preceq' y, \\ (a, x) \approx'' (b, y) &\iff (a \approx b \Rightarrow x \approx' y) \end{aligned}$$

It is standard to check that $\mathcal{S}(W \Rightarrow W')$ is isomorphic to the space $\mathcal{S}(W) \rightarrow \mathcal{S}(W')$ of continuous functions from $\mathcal{S}(W)$ to $\mathcal{S}(W')$. The intuition behind the conditions above is that the pairs (a, x) encode the step functions $\varepsilon_{\downarrow a, \downarrow x}$, which are the prime elements of the latter domain. The isomorphism is realized by :

$$\begin{aligned} Tr_{W, W'}(f) &:= \left\{ (a, x) \in D_f^{\hat{\approx}} \times D' / x \in f(\downarrow a) \right\}, \\ \text{where "Tr"} &\text{ abbreviates "trace", and} \\ Tr_{W, W'}^{-1}(T)(v) &:= \left\{ x \in D' / \exists a \in D_f^{\hat{\approx}} a \subseteq v \text{ and } (a, x) \in T \right\} \\ \text{where } f \in \mathcal{S}(W) \rightarrow \mathcal{S}(W'), v \in \mathcal{S}(W) &\text{ and } T \in \mathcal{S}(W \Rightarrow W'). \end{aligned}$$

Remark 4 " \preceq'' ", as defined above, is a preorder and not a partial order, even if \preceq and \preceq' are partial orders. That is the reason why preorders have been considered from the beginning.

3.5 Retraction pairs for application and abstraction.

Suppose there is a morphism j from $W \Rightarrow W'$ to W'' , then $\mathcal{S}(W) \rightarrow \mathcal{S}(W')$ is a retract of $\mathcal{S}(W'')$ under the retraction pair (q_j, ap_j) defined by $q_j := \mathcal{S}(j) \circ Tr_{W, W'}$ and $ap_j := Tr_{W, W'}^{-1} \circ \mathcal{P}(j)$. Of course j depends on W, W', W'' . One has to note that our notation is slightly ambiguous, since the same map j can

¹⁴Event structures are issued from work of G. Kahn and G. Plotkin on the sequentiality of λ -calculus. They are used for modelling processes. We are grateful to G. Winskel for useful pointers to the basic literature on event structures.

¹⁵The present definition of morphisms arises from the definition of continuous models of untyped λ -calculus given in [18] in a more restricted context (such models are called K -models in section 5).

happen to be a morphism relatively to distinct triples, and then will give rise to different pairs (q_j, ap_j) ¹⁶. We keep it however since it is very convenient, and will manage ambiguous occurrences when needed ¹⁷.

For further practical uses we make explicit the conditions on j , as well as the action of the maps ap_j and q_j on their arguments. Thus :

$$\begin{aligned} j & \text{ is an injective map from } D_f^\diamond \times D' \text{ into } D'' \text{ satisfying :} \\ j(a, x) \preceq j(b, y) & \implies b \preceq a \text{ and } x \preceq' y \quad (*) \\ j(a, x) \diamond j(b, y) & \iff (a \diamond b \implies x \diamond' y) \quad (**) \end{aligned}$$

and the values $q_j(f)$ and $ap_j(u)(v)$, for $f \in \mathcal{S}(W) \rightarrow \mathcal{S}(W')$, $u \in \mathcal{S}(W'')$, $v \in \mathcal{S}(W)$ are given by :

$$\begin{aligned} ap_j(u)(v) & := \downarrow' \left\{ x \in D / \exists a \in D_f^\diamond (a \subseteq v \text{ and } j(a, x) \in u) \right\} \text{ and} \\ q_j(f) & := \downarrow'' \left\{ j(a, x) / a \in D_f^\diamond, x \in f(\downarrow a) \right\}. \end{aligned}$$

Alternatively one could check directly from these definitions that ap_j and q_j are continuous, have the right ranges, and that they satisfy $ap_j \circ q_j = id$, thus making precise the role of $(*)$ and $(**)$ ¹⁸.

Remark 5 *The external “ \downarrow ” in the definition of q or ap can be redundant in some cases, and then need not be mentioned. This happens obviously for q when the range of j is \downarrow -closed, and in particular when j is onto or \preceq is equality (case of \mathcal{E}^2); it happens for ap if j is an embedding. This remark will be used in the examples. If the preorder is trivial then the internal arrows should also be dropped.*

3.6 Reflexive webs and webbed models of untyped λ -calculus.

The category of prime webs admits reflexive objects, namely webs W such that there is a morphism j from $W \Rightarrow W$ to W . This is a particular case of the preceding subsection. Then $M := (\mathcal{S}(W), q_j, ap_j)$ is a reflexive object of $Univ$, i.e. a model of untyped λ -calculus.

A *reflexive web* is hence a tuple $(W, j) = (D, \diamond, \preceq, j)$ where j is an injective map from $D_f^\diamond \times D$ into D satisfying :

$$\begin{aligned} j(a, x) \preceq j(b, y) & \implies b \preceq a \text{ and } x \preceq y \quad (*) \\ j(a, x) \diamond j(b, y) & \iff (a \diamond b \implies x \diamond y) \quad (**) \end{aligned}$$

The model is extensional iff j is an isomorphism.

In particular a trivial preorder can never give rise to an extensional model ¹⁹.

¹⁶This happens since it is possible in some cases to vary \diamond' and \diamond'' and however keep the same j . In this case the domains and ranges of q_j and ap_j , will be changed, and hence q_j and ap_j will be changed too, even if their formal definition remains (superficially) the same.

¹⁷This remark indeed concerns all the j_{hom} that we will define later on, and which will give rise, in particular, to two fundamental retraction pairs, namely (Q, AP) and $(\Lambda, Appl)$.

¹⁸First $(*)$ forces $ap \circ q = id$, second $(*)$ plus the direction \implies of $(**)$ force $ap(u)(v)$ to be a clique ; finally, the direction \iff of $(**)$ forces $q(f)$ to be a clique.

¹⁹This is not true in the stable semantics (see for example [16] or [19]).

4 Webbed models for F .

We start from a web of the form :

$$(\Omega, \approx_{hom}, \approx_{coh}, \preceq_{hom}, \preceq_{coh}, j_{hom}, j_{hcoh})$$

where Ω is a set, followed by two reflexive and symmetric relations, two pre-orders, and two functions.

The relations \approx_{hom} and \approx_{coh} are called respectively *homogeneity* and *coherence*. We say that $a \subseteq \Omega$ is *homogeneous* (resp. *coherent*) if its elements are pairwise homogeneous (resp. coherent). We let Ω^{hom} and Ω_f^{hom} denote the set of homogeneous (resp. finite and homogeneous) subsets of Ω . We let *hcoh* abbreviate “homogeneous and coherent”, for example Ω^{hcoh} and Ω_f^{hcoh} denote respectively the set of homogeneous and coherent subsets of Ω , and the set of the finite ones, and similarly for D^{hcoh} and D_f^{hcoh} if $D \subseteq \Omega$, also $\approx_{hcoh} := \approx_{hom} \cap \approx_{coh}$. Finally j_{hom} is an injection from $\Omega_f^{hom} \times \Omega$ to Ω , and j_{hcoh} an injection from $\Omega_f^{hcoh} \times \Omega$ to Ω .

Notations. In the following H denotes, for short, the set of homogeneous and \downarrow_{hom} -closed subsets D of Ω , and for any relation r on Ω , r^D will denote the restriction of r to D . We let :

$$\begin{aligned} W_{hom} &:= (\Omega, \approx_{hom}, \preceq_{hom}). \text{ Thus } \mathcal{S}(W_{hom}) = (H, \subseteq) \\ W_{coh} &:= (\Omega, \approx_{coh}, \preceq_{coh}) \\ W_{hhcoh} &:= (\Omega, \approx_{hcoh}, \preceq_{hom}). \text{ Finally, for all } D \in H \text{ we let:} \\ W_D &:= (D, \approx_{coh}^D, \preceq_{coh}^D) = (D, \approx_{hcoh}^D, \preceq_{coh}^D). \end{aligned}$$

We will now introduce step by step the constants and constructors we need, together with the conditions which make things work. At the end we will have reached a set of 11 conditions, which generalize the three ones which were needed to have a model of untyped λ -calculus (cf. Section 3.6). These conditions may look rather technical at first sight but they are quite natural in the light of the preliminaries of Section 3. *They are furthermore easy to fulfill*, as show the examples in Section 5 (the reader might have a look at these models before proceeding further).

4.1 Definition of *Types*, *types*, Q and AP .

Types will be a prime algebraic Scott domain, whose elements are prime algebraic Scott domains. *Types* will be isomorphic to some $\mathcal{S}(W)$ called *types*, and hence will be prime algebraic, but *Types* is not of the form $\mathcal{S}(W')$.

The first requirements express that (W_{hom}, j_{hom}) is a reflexive prime web and that W_{coh} is a prime web. This implies that W_D is a prime web and a subweb of W_{coh} , for each $D \in H$, and that W_D is a subweb of $W_{D'}$ iff $D \subseteq D'$. We let $X_D := \mathcal{S}(W_D)$.

We then define *types* := $\mathcal{S}(W_{hom}) = (H, \subseteq)$.

We then define $Types$ as the set of all X_D , $D \in H$.

It is easy to check that the correspondence between $types$ and $Types$ is 1-1.

Since (W_{hom}, j_{hom}) is a reflexive web, $(types, Q, AP)$ is a reflexive domain, where $AP := ap_{j_{hom}}$ and $Q := q_{j_{hom}}$, as defined in section 3.6 (the explicit definition of Q is recalled in the footnote ²⁰ below). In particular Q is left invertible : $AP \circ Q = id_{types \rightarrow types}$.

Written explicitly, the requirements are :

$$\preceq_{hom} \text{ and } \simeq_{hom} \text{ (resp. } \preceq_{coh} \text{ and } \simeq_{coh} \text{) are compatible} \quad (1)$$

and, for all $x, y \in \Omega$ and $a, b \in \Omega_f^{hom}$:

$$(j_{hom}(a, x) \preceq_{hom} j_{hom}(b, y)) \implies (b \preceq_{hom} a \text{ and } x \preceq_{hom} y) \quad (2)$$

$$(j_{hom}(a, x) \simeq_{hom} j_{hom}(b, y)) \iff (b \simeq_{hom} a \implies x \simeq_{hom} y) \quad (3)$$

4.2 Definition of $Terms$.

The *next requirement* ²¹ is :

$$\preceq_{hom} \subseteq \preceq_{coh} \quad (4)$$

Under this condition W_{hhcoh} is a prime web and we define :
 $Terms := \mathcal{S}(W_{hhcoh})$.

Lemma 8 1. $Terms \subseteq types$

2. $Terms = \cup Types$

3. Each $X \in Types$ is a substructure of $Terms$.

Proof. The first assertion is immediate. For the second one we have to show that a subset a of Ω belongs to some X_D iff $a \in \Omega^{hhcoh}$ and $\downarrow_{hom} a = a$. Suppose $a \in \mathcal{S}(W_D)$, then a is homogeneous and coherent ; if $x \preceq_{hom} y \in a$ then $x \in D$ since D is \downarrow_{hom} -closed, and $x \in a$ since $\preceq_{hom} \subseteq \preceq_{coh}$ and a is \downarrow_{coh}^D -closed. Conversely, suppose that a is homogeneous, coherent, and \downarrow_{hom} -closed, then $a \in H$ and, trivially, a is closed under \downarrow_{coh}^a , hence $a \in \mathcal{S}(W_a)$. The third claim is then obvious since X and $Terms$ are both ordered by inclusion, $\emptyset \neq X \subseteq Terms$, and since X is closed under directed unions.

²⁰ $Q(F) := \downarrow_{hom} \left\{ j_{hom}(a, x) / (a, x) \in \Omega_f^{hom} \times \Omega \text{ and } x \in F(\downarrow_{hom}(a)) \right\}$

²¹If none of the preorders were included in the other one, then we should have to use a variant S° of S to build $Term$ out of its prime web (cf. Section 7.3). Moreover the definition of the various operators is more delicate in the general setting, and might even raise real problems.

We made the choice $\preceq_{hom} \subseteq \preceq_{coh}$ above since this property was true in the BB -model.

By contrast, one can note that there is no technical advantage to ask for $\simeq_{coh} \subseteq \simeq_{hom}$, even if this would be natural from an intuitive point of view.

4.3 Definition of \Rightarrow .

We define \Rightarrow on *types* by :

$$D \Rightarrow D' := \downarrow_{hom} \left\{ j_{hcoh}(a, x) / (a, x) \in D_f^{hcoh} \times D' \right\}.$$

To ensure that this is an element of *types* it is enough to require that :
for all $x, y \in \Omega$ and $a, b \in \Omega_f^{hcoh}$:

$$j_{hcoh}(a, x) \simeq_{hom} j_{hcoh}(b, y) \iff (a \simeq_{hom} b \text{ and } x \simeq_{hom} y) \quad (5)$$

The continuity of \Rightarrow in both arguments is then clear.

One can observe that $D \Rightarrow D'$ is the natural representative of $D \ni D'$ within Ω and as an element of *types* : we have $D \Rightarrow D' = \downarrow_{hom} j_{hcoh}^\bullet(D \ni D')$ and j_{hcoh} is a morphism from $W_{D \ni D'}$ to $W_{D \Rightarrow D'}$.

4.4 Definition of *apl* and *lbd*.

We now add :

$$j_{hcoh}(a, x) \preceq_{coh} j_{hcoh}(b, y) \implies (b \preceq_{coh} a \text{ and } x \preceq_{coh} y) \quad (6)$$

$$j_{hcoh}(a, x) \simeq_{coh} j_{hcoh}(b, y) \iff (b \simeq_{coh} a \implies x \simeq_{coh} y) \quad (7)$$

These conditions express that the restriction $j_{hcoh}^{D, D'}$ of j_{hcoh} to $D_f^{hcoh} \times D'$ is a morphism from $W_D \ni W_{D'}$ to $W_{D \Rightarrow D'}$, hence $X_D \rightarrow X_{D'}$ is a retract of $X_{D \Rightarrow D'}$. The explicit definition of (*lbd*, *apl*), which will be used in examples, is given in the footnote ²² below.

To have an isomorphism, as required for extensional models, we have to replace (6) by the two following conditions :

$$j_{hcoh}(a, x) \preceq_{coh} j_{hcoh}(b, y) \iff (b \preceq_{coh} a \text{ and } x \preceq_{coh} y) \quad (\text{ext-1})$$

and, for all $z \in \Omega$ and $(a, x) \in \Omega_f^{hcoh} \times \Omega$:

$$z \preceq_{hom} j_{hcoh}(a, x) \implies \exists (a', x') \in \Omega_f^{hcoh} \times \Omega \quad z \sim_{coh} j_{hcoh}(a', x') \quad (\text{ext-2})$$

4.5 Definition of *Appl* and *Lambda*.

The next condition says that, for all $x, y \in \Omega$ and $a, b \in \Omega_f^{hom}$:

$$j_{hom}(a, x) \simeq_{coh} j_{hom}(b, y) \iff (a \simeq_{hom} b \implies x \simeq_{coh} y) \quad (8)$$

$$j_{hom}(a, x) \preceq_{coh} j_{hom}(b, y) \implies (b \preceq_{hom} a \text{ and } x \preceq_{coh} y) \quad (9)$$

²²For $D, D' \in H$, $f : X_D \rightarrow X_{D'}$, $u \in X_D$, $T \in X_{D \Rightarrow D'}$.

$lbd_{D, D'}(f) = \downarrow_{coh}^{D \Rightarrow D'} \left\{ j_{hcoh}(a, x) / (a, x) \in D_f^{hcoh} \times D' \text{ and } x \in f(\downarrow_{coh}^D(a)) \right\}$
 $apl_{D, D'}(T, u) := \downarrow_{coh}^{D'} \left\{ x / \exists a \in D_f^{hcoh} \quad a \subseteq u \text{ and } j_{hcoh}(a, x) \in T \right\}.$

Conditions (8, 2, 3), imply that j_{hom} is a morphism from $W_{hom} \Rightarrow W_{hhcoh}$ to W_{hhcoh} , hence $types \rightarrow Terms$ is a retract of $Terms$, via the retraction pair $(\text{Lambda}, \text{Appl})$ generated by j_{hom} in this context.

It is useful here to give the explicit definition of $Q, \text{Lambda}, \text{Appl}$:
For all $F \in \text{Hom}(types, types)$, $f \in \text{Hom}(types, Terms)$, $D \in H$ and $u \in Terms$:

$$\begin{aligned} Q(F) &:= \downarrow_{hom} \{ j_{hom}(a, x) / (a, x) \in \Omega_f^{hom} \times \Omega \text{ and } x \in F(\downarrow_{hom} a) \} \\ \text{Lambda}(f) &:= \downarrow_{hom} \{ j_{hom}(a, x) / (a, x) \in \Omega_f^{hom} \times \Omega \text{ and } x \in f(\downarrow_{hom} a) \} \\ \text{Appl}(u)(D) &:= \downarrow_{hom} \{ x \in \Omega / \exists a \in \Omega_f^{hom} a \subseteq D \text{ and } j_{hom}(a, x) \in u \} \end{aligned}$$

Now, there only remains to add two conditions which, together with (9) above, will allow us to set up the links between $\text{Hom}_F(types, Terms)$ and $Q(F)$, for $F \in \text{Hom}(types, types)$. These two conditions are less easy to justify, and condition (11) looks rather complicated at first sight. So it is interesting to have in mind that *they are automatically fulfilled if \preceq_{coh} is trivial* (then \preceq_{hom} is trivial also) and that (11) is also fulfilled when the converse of (9) holds (these two particular instances will cover all the examples given in Section 5).

$$z \preceq_{hom} j_{hom}(a, x) \implies \exists (a', x') \in \Omega_f^{hom} \times \Omega \quad z \sim_{hom} j_{hom}(a', x') \quad (10)$$

For $x, y \in \Omega$, $F \in \text{Hom}(types, types)$, $D \in types$, $b \in D_f$:

$$\left\{ \begin{array}{l} x \preceq_{coh} y \\ x \in F(D) \\ y \in F(\downarrow_{hom} b) \end{array} \right\} \implies \exists a \in D_f \left\{ \begin{array}{l} j_{hom}(a, x) \preceq_{coh} j_{hom}(b, y) \\ x \in F(\downarrow_{hom} a) \end{array} \right\} \quad (11)$$

Recall that $\text{Hom}_F(types, Terms)$ is the set consisting of the morphisms $f \in \text{Hom}(types, Terms)$ such that $f(D) \in X_{F(D)}$ for all D . It is easy to see that $\text{Hom}_F(types, Terms)$ is the underlying set of a subdomain of $types \rightarrow Terms$, that we will denote by $types \rightarrow_F Terms$ (and which is also a prime algebraic domain).

Proposition 9 *Conditions (3,4,9,10,11) imply that:*

1. *If $f \in \text{Hom}_F(types, Terms)$, then $\text{Lambda}(f) \in X_{Q(F)}$*
2. *If $D \in types$ and $u \in X_{Q(F)}$, then $\text{Appl}(u)(D) \in X_{F(D)}$*

Thus Lambda and Appl induce a retraction pair $(\text{Lambda}_F, \text{Appl}_F)$ between $types \rightarrow_F Terms$ and $X_{Q(F)}$.

Before we start the proof let us notice that it is easy to show, from (3) and from the monotonicity of F , that for all $a \in \Omega_f^{hom}$ and $x \in \Omega$ we have:

$$j_{hom}(a, x) \in Q(F) \implies x \in F(\downarrow_{hom} a) \quad (*)$$

Proof of Proposition 9. The proof of the first assertion uses furthermore (9,10,4) and that of the second uses (4,11).

1. We already know that $\text{Lambda}(f)$ is homogeneous and coherent, since it is an element of Terms . There remains to show that $\text{Lambda}(f)$ is a subset of $Q(F)$ and that it is closed under $\downarrow_{\text{coh}}^{Q(F)}$. The first assertion is clear from the definition of $\text{Lambda}(f)$, using that $f(D) \subseteq F(D)$ for all homogeneous D (since $f \in \text{Hom}_F(\text{types}, \text{Terms})$). Suppose now that $z \in Q(F) \cap \downarrow_{\text{coh}} \text{Lambda}(f)$. We want to show that $z \in \text{Lambda}(f)$. From the definition of $Q(F)$ there exist a, x such that $z \preceq_{\text{hom}} j_{\text{hom}}(a, x)$ and $x \in F(\downarrow_{\text{hom}} a)$. From (10) we have $z \sim_{\text{hom}} j_{\text{hom}}(a', x')$. Since $Q(F)$ is \downarrow_{hom} -closed, $j_{\text{hom}}(a', x') \in Q(F)$, and by (*) $x' \in F(\downarrow_{\text{hom}} a')$. Let $z' \in \text{Lambda}(f)$ be such that $z \preceq_{\text{coh}} z'$, and let (c, s) be such that $z' \preceq_{\text{hom}} j_{\text{hom}}(c, s)$ and $s \in f(\downarrow_{\text{hom}} c)$. Now we have $j_{\text{hom}}(a', x') \sim_{\text{hom}} z \preceq_{\text{coh}} z' \preceq_{\text{hom}} j_{\text{hom}}(c, s)$. By (4) and the transitivity of \preceq_{coh} we have $j_{\text{hom}}(a', x') \preceq_{\text{coh}} j_{\text{hom}}(c, s)$, hence, by (9), $x' \preceq_{\text{coh}} s$ and $c \preceq_{\text{hom}} a'$. Now, $f(\downarrow_{\text{hom}} c) \in F(\downarrow_{\text{hom}} c)$ since $f \in \text{Hom}_F(\text{types}, \text{Terms})$, so $f(\downarrow_{\text{hom}} c)$ is closed under $\downarrow_{\text{coh}}^{F(\downarrow_{\text{hom}} c)}$. Hence $x \in f(\downarrow_{\text{hom}} c) \subseteq f(\downarrow_{\text{hom}} a')$. Thus $j_{\text{hom}}(a', x') \in \text{Lambda}(f)$ and $z \sim_{\text{hom}} j_{\text{hom}}(a', x')$ also, q.e.d.
2. We already know that $\text{Appl}(u)(D) \in \text{Terms}$. So $\text{Appl}(u)(D)$ is homogeneous, coherent, and \downarrow_{hom} -closed. There remains to see that it is contained in $F(D)$, and is $\downarrow_{\text{coh}}^{F(D)}$ -closed. Suppose $y \in \text{Appl}(u)(D)$. Then $y \preceq_{\text{hom}} x$ for some x such that $j_{\text{hom}}(a, x) \in u$ for some $a \subseteq D$. Now, $u \subseteq Q(F)$, thus $j_{\text{hom}}(a, x) \in Q(F)$, hence $x \in F(\downarrow_{\text{hom}} a)$, by (*); so $x \in F(D)$. Suppose now that $y \in \text{Appl}(u)(D)$ and $x \in F(D)$ and $x \preceq_{\text{coh}} y$. Since $\preceq_{\text{hom}} \subseteq \preceq_{\text{coh}}$ by (4) we can assume w.l.o.g. that $j_{\text{hom}}(b, y) \in u$. Since $u \in Q(F)$ we have $y \in F(\downarrow_{\text{hom}} b)$ by (*). Now, $u \subseteq Q(D)$; by (11) there is an $a \in D_f$ such that $j_{\text{hom}}(a, x) \preceq_{\text{coh}} j_{\text{hom}}(b, y)$ and $x \in F(\downarrow_{\text{hom}} a)$. So $j_{\text{hom}}(a, x) \in Q(F)$. Since u is $\downarrow_{\text{coh}}^{Q(F)}$ -closed and $j_{\text{hom}}(b, y) \in u$, we have $j_{\text{hom}}(a, x) \in u$. Hence $x \in \text{Appl}(u)(D)$, q.e.d.

The retraction pairs $(\text{Lambda}_F, \text{Appl}_F)$ are pairs of inverse isomorphisms if the two conditions (9,11) are replaced by the following one condition :

$$j_{\text{hom}}(a, x) \preceq_{\text{coh}} j_{\text{hom}}(b, y) \iff (b \preceq_{\text{hom}} a \text{ and } x \preceq_{\text{coh}} y) \quad (\text{ext-3})$$

(according to Section 3.2 one would expect two conditions here, but the second one is nothing else than (10)).

Note that (ext-3) implies trivially (11) since in this case it is easy to check that $a = b \cup \{x\}$ is a solution.

4.6 Conclusion.

A web which satisfies the eleven conditions numbered (1) to (11) in the section above gives rise to a model of F , which belongs to the continuous semantics and is a polymax model in the sense of Section 2.1.

A web which satisfies the eleven conditions obtained by replacing (6,9,11) by the stronger set of conditions (ext-1,ext-2,ext-3) gives rise to an extensional model of F .

5 Examples.

We present now a few simple and less simple examples. We must confess that, at present, we know nearly nothing about the equational theories of these examples (which does not mean that one can say nothing about them). The only thing we can assert is whether they are extensional or not, since we control this point when building them. In other words, although our class contains many models, satisfying a great variety of domain equations, we do not know whether they are essentially different at the level of term-equations (between pure F -terms) or not.

What we know is that *the class contains many complete models* for $F\eta$, since many of its models will satisfy the conditions in [5]. The class could also contain other kinds of extensional complete models, since the conditions in [5] are only sufficient conditions, a priori. At this stage of our present knowledge it could even be the case that all the models of the class are complete for F or for $F\eta$. This is a very drastic conjecture, which is probably false. We make weaker ones, presenting in particular a candidate for β -completeness. This model, which is a simplification of the BB -model, fulfills all the conditions given in [5] but is not extensional.

The interest of trying to answer this latter conjecture is that, whatever the answer will be, it will force us to better understand completeness, and hopefully to find less technical conditions than the ones which are proposed in [5] (in the same sense that Simpson's paper [34] is progress with respect to Friedman's one [13]).

5.1 The square models.

Definition 10 A square-model is a webbed model of the form $(\Omega, \simeq, \simeq, \preceq, \preceq, j, j)$, where $M := (\Omega, \simeq, \preceq, j)$ is a reflexive prime web. A necessary and sufficient condition for a reflexive prime web M to give rise to a square-model M^2 of F is that Conditions (10,11) hold, since all other conditions are immediately fulfilled.

Remark 6 In a square model $types = Terms$ and $Q = Lambda$. The square models with trivial coherence are exactly those webbed models of F which are (implicit) models of the calculus of constructions plus "types = kinds" [6].

Remark 7 If M is extensional, then the two conditions (10,11) hold and M^2 is extensional ; moreover M^2 is extensional only if M is. As noted previously another case where the conditions hold is when \preceq is trivial.

5.1.1 The graph²-models.

A graph-model of untyped λ -calculus is a reflexive prime web of the form $W := (\Omega, j)$, these models are also called Plotkin-Scott-Engeler's algebras in the literature, they are complete lattices, and the most well-known of them are

Plotkin-Scott's P_ω and Engeler's \mathcal{E} , which is recalled below. Thanks to the remark above each graph model gives rise to a square-model.

5.1.2 The \mathcal{E}^2 -model.

The simplest model of untyped λ -calculus is Engeler's and Plotkin's graph model $\mathcal{E} := (\Omega, j)$, where Ω is the least solution of $\Omega = A \cup (\Omega_f \times \Omega)$, and j is the identity injection (inclusion). Here A is a nonempty set of "atoms", namely of elements of the underlying set-theoretical universe which are not pairs. Then \mathcal{E}^2 is a nonextensional model of F , satisfying :

$$\text{types} = \text{Terms} = (P(\Omega), \subseteq), \text{Types} = (\{P(D) \mid D \subseteq \Omega\}, \subseteq).$$

The interpretation of F -types and F -terms in the \mathcal{E}^2 -model goes as follows, using Remark 5 in Section 3.5, which allows dropping the arrows.

Here a always ranges over Ω_f , and x over Ω .

$$\begin{aligned} |\alpha|_\rho &\in P(\Omega) \\ |\sigma \rightarrow \tau|_\rho &= \left\{ (a, x) \mid a \subseteq |\sigma|_\rho \text{ and } x \in |\tau|_\rho \right\} \\ |\forall \alpha. \sigma|_\rho &= \left\{ (a, x) \mid x \in |\sigma|_{\rho[\alpha:=a]} \right\} \\ |x^\sigma|_\rho &\in P(|\sigma|_\rho) \\ |t^{\sigma \rightarrow \tau} u^\sigma|_\rho &= \left\{ x \mid \exists a \subseteq |u|_\rho \ (a, x) \in |t^{\sigma \rightarrow \tau}|_\rho \right\} \\ |\lambda x^\sigma. t^\tau|_\rho &= \left\{ (a, x) \mid a \subseteq |\sigma|_\rho \text{ and } x \in |t^\tau|_{\rho[x^\sigma:=a]} \right\} \\ |t^{\forall \alpha. \sigma} \tau|_\rho &= \left\{ x \mid \exists a \subseteq |\tau|_\rho \text{ and } (a, x) \in |t^{\forall \alpha. \sigma}|_\rho \right\} \\ |\lambda \alpha. t^\sigma|_\rho &= \left\{ (a, x) \mid x \in |t^\sigma|_{\rho[\alpha:=a]} \right\} \end{aligned}$$

Examples :

$$\begin{aligned} |\forall \alpha. \alpha|_\rho &= \{ (a, x) \mid x \in a \} \\ |\forall \alpha. \alpha \rightarrow \alpha|_\rho &= \{ (a, (b, x)) \mid b \subseteq a \text{ and } x \in a \} \\ |\lambda x^\sigma. x^\sigma|_\rho &= \{ (a, x) \mid a \subseteq |\sigma|_\rho \text{ and } x \in a \} \\ |\lambda \alpha. \lambda x^\alpha. x^\alpha|_\rho &= \{ (a, (b, x)) \mid b \subseteq a \text{ and } x \in b \} \\ |\lambda y^\sigma. \lambda x^\tau. x^\tau|_\rho &= \{ (a, (b, x)) \mid a \subseteq |\sigma|_\rho, b \subseteq |\tau|_\rho \text{ and } x \in b \} \end{aligned}$$

Remark 8 *If A is infinite, then \mathcal{E}^2 is a model of F extended with products..*

*Hint ²³: start from a partition of A into two infinite disjoint sets A_1 and A_2 and from two bijections ϕ_1, ϕ_2 between A and A_1 and A and A_2 . Extend these bijections to Ω via $\bar{\phi}_i(a, x) := (a, \phi_i(x))$. Define Ω_i as the range of $\bar{\phi}_i$. Then Ω is the disjoint union of Ω_1 and Ω_2 . We finally define $D_1 * D_2$ as $\bar{\phi}_1^\bullet(D_1) \cup \bar{\phi}_2^\bullet(D_2)$. This gives rise to an isomorphism $*$ between types \times types and types, and clearly we have $P(D_1 * D_2)$ isomorphic to $P(D_1) \times P(D_2)$.*

The proof only depends on a "symmetry" property of the web of \mathcal{E}^2 (or simply \mathcal{E}), which can also be directly found in a lot of other models (but not all), or forced voluntarily during the construction of a model. The way for modelling

²³This is similar to a proof given for Scott's D_∞ in [18], which dates back to Scott.

other constructs can call for more complex webs, but the basic principle is the same.

5.1.3 The (ext-K)²-models.

A *K-model* is a reflexive prime web of the form $W := (\Omega, \preceq, j)$. *K-models* were isolated in [18], and are prime algebraic complete lattices. The family includes in particular Scott's and Park's D_∞ -models. If K is an extensional \mathcal{K} -model, then \mathcal{K}^2 is an extensional model of F . This is a particular case of the remark which follows the definition of square models.

Here $types = Terms = (S^\perp(\Omega), \subseteq)$, $Types : (\{S^\perp(D) / D \in S^\perp(\Omega)\}, \subseteq)$.

5.2 The *BB*-model.

Relations \simeq are usually introduced in the continuous semantics to produce solutions of domain equations which have no solution among complete lattices.

This is the case for the *BB*-model built in [2], whose construction is recalled below. As already mentioned this model is complete for System F_η (cf. [5]). Here the web is such that all relations and functions are distinct and nontrivial.

The construction goes as follows ; we divide it into two steps. In the next example we will drop the second step, taking then the preorders as trivial. We will then get a simpler model which will enjoy all the main properties of the *BB*-model, except extensionality.

First step: definition of Ω , of the coherence relations, and of the two injections "j". We fix a countable set of atoms $A = N \cup L$, with N and L infinite, and we fix $\varepsilon \in A$. The elements of A are supposed not to be pairs or triples. We define two increasing sequences of webs $(\Omega_n, \simeq_{hom}^n)$ and $(\Omega_n, \simeq_{coh}^n)$, by induction on n , and two increasing sequences of injections $j_{hom}^n : \Omega_{n,f}^{hom} \times \Omega_n \rightarrow \Omega_{n+1}$ and $j_{hcoh}^n : \Omega_{n,f}^{hcoh} \times \Omega_n \rightarrow \Omega_{n+1}$ as follows :

$\Omega_0 = A$, the elements of A are pairwise incoherent, the elements of L are pairwise non-homogeneous, and all other possible pairs of atoms are homogeneous. here "pairwise" refer to pairs of distinct elements, in particular \simeq_{hom}^0 and \simeq_{coh}^0 are reflexive.

$\Omega_{n+1} = A \cup (\Omega_{n,f}^{hom} \times \Omega_n \times \{\varepsilon\}) \cup (\Omega_{n,f}^{hcoh} \times \Omega_n)$ (this is a disjoint union)

$j_{hom}^n(a, x) := (a, x, \varepsilon)$, abbreviated as $\langle a, x \rangle$.

j_{hcoh}^n is just the identity injection.²⁴

Finally, we define the relations on Ω_{n+1} . If $x', y' \in \Omega_{n+1}$ are situated in two different components, then they are neither homogeneous nor coherent. If $x', y' \in A$, then they are related as in Ω_0 . We now consider the remaining cases. For $x, y \in \Omega_n$ and $a, b \in \Omega_{n,f}^{hom}$ (for $\langle -, - \rangle$) or $a, b \in \Omega_{n,f}^{hcoh}$ (for $(-, -)$), we let :

²⁴A variant, which gives a slightly simpler domain equation for *types* (and is closer to the version in [2]) is to take $A := N \times L$ and to define $(n, l) \simeq_{hom} (n', l')$ iff $l = l'$, and $j_{hcoh}^n(a, x) := (a, x, \varepsilon, \varepsilon)$ instead of (a, x) , everything else being unchanged.

$$\begin{aligned}
\langle a, x \rangle \approx_{hom}^{n+1} \langle b, y \rangle &\iff (a \approx_{hom}^n b \Rightarrow x \approx_{hom}^n y) \\
(a, x) \approx_{hom}^{n+1} (b, y) &\iff (a \approx_{hom}^n b \text{ and } x \approx_{hom}^n y) \\
(a, x) \approx_{coh}^{n+1} (b, y) &\iff (a \approx_{coh}^n b \Rightarrow x \approx_{coh}^n y) \\
\langle a, x \rangle \approx_{coh}^{n+1} \langle b, y \rangle &\iff (a \approx_{hom}^n b \Rightarrow x \approx_{coh}^n y)
\end{aligned}$$

Then $(\Omega, \approx_{hom}, \approx_{coh}, j_{hom}, j_{hcoh})$ is taken as the limit, in the obvious sense, of the obviously increasing sequence above.

Second step : definition of the preorders. These relations, which will happen to be preorders at the end, are defined as: reflexive, trivial on A , not relating any two elements which belong to two different components of Ω ; elsewhere they are defined by induction on n :

$$\begin{aligned}
\langle a, x \rangle \preceq_{hom}^{n+1} \langle b, y \rangle &\iff b \preceq_{hom}^n a \text{ and } x \preceq_{hom}^n y \\
(a, x) \preceq_{hom}^{n+1} (b, y) &\iff (a, x) = (b, y) \\
(a, x) \preceq_{coh}^{n+1} (b, y) &\iff b \preceq_{coh}^n a \text{ and } x \preceq_{coh}^n y \\
\langle a, x \rangle \preceq_{coh}^{n+1} \langle b, y \rangle &\iff b \preceq_{hom}^n a \text{ and } x \preceq_{coh}^n y
\end{aligned}$$

Then, $(\Omega, \approx_{hom}, \preceq_{hom}, \approx_{coh}, \preceq_{coh}, j_{hom}, j_{hcoh})$ is easily checked to be an extensional model for F .

We can note that at the end we have :

$$\Omega = A \cup (\Omega_f^{hom} \times \Omega \times \{\varepsilon\}) \cup (\Omega_f^{hcoh} \times \Omega).$$

and that, moreover, if x, y belong to two different components of Ω , then x and y are neither homogeneous nor coherent. This recursive equation on Ω corresponds to a subtle recursive domain equation satisfied by *types* :

$$types \simeq (\cup_{l \in L} (P(N \cup \{l\}))) \oplus (types \rightarrow types) \oplus \left(\bigcup_{D, D' \in types} P(D \Rightarrow D') \right)$$

where \oplus means that we are taking the disjoint union of the three domains, except for their bottom elements, which are amalgamated into a single one. The existence and structure of the first component of *types*, (namely that it contains an infinite number of pairwise disjoint and infinite flat domains), together with the trichotomy of *types*, allows to define the suitable morphisms C , *index*, and *case*, which ensure the satisfaction of the two last conditions for $\beta\eta$ -completeness given in .

Remark 9 *The model satisfies : $x \preceq_{hom} y \Rightarrow x \preceq_{coh} y \Rightarrow x \approx_{hom} y$.*

The interpretation of F -types and F -terms now goes as follows, using once more Remark 5 in Section 3.5.

Here a always ranges over Ω_f^{hom} or Ω_f^{hcoh} , and x over Ω .

$|\alpha|_\rho$ is a homogeneous and \downarrow_{hom} -closed subset of Ω .

$$\begin{aligned} |\sigma \rightarrow \tau|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in |\tau|_\rho \right\} \\ |\forall \alpha. \sigma|_\rho &= \left\{ \langle a, x \rangle / x \in |\sigma|_{\rho[\alpha := \downarrow_{hom} a]} \right\}. \end{aligned}$$

$$\begin{aligned} |x^\sigma|_\rho &\text{ is homogeneous and coherent, and is an } \downarrow_{coh}^{|\sigma|_\rho}\text{-closed subset of } |\sigma|_\rho. \\ |t^{\sigma \rightarrow \tau} u^\sigma|_\rho &= \left\{ x / \exists a \subseteq |u|_\rho \ (a, x) \in |t^{\sigma \rightarrow \tau}|_\rho \right\} \\ |\lambda x^\sigma. t^\tau|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in |t^\tau|_{\rho[x^\sigma := \downarrow_{coh}^{|\sigma|_\rho} a]} \right\} \\ |t^{\forall \alpha. \sigma} \tau|_\rho &= \left\{ x / \exists a \subseteq |\tau|_\rho \text{ and } \langle a, x \rangle \in |t^{\forall \alpha. \sigma}|_\rho \right\} \\ |\lambda \alpha. t^\sigma|_\rho &= \left\{ \langle a, x \rangle / x \in |t|_{\rho[\alpha := \downarrow_{hom} a]} \right\} \end{aligned}$$

Examples :

$$\begin{aligned} |\forall \alpha. \alpha|_\rho &= \left\{ \langle a, x \rangle / x \in \downarrow_{hom} a \right\} \\ |\forall \alpha. \alpha \rightarrow \alpha|_\rho &= \left\{ \langle a, (b, x) \rangle / b \subseteq \downarrow_{hom} a \text{ and } x \in \downarrow_{hom} a \right\} \\ |\lambda x^\sigma. x^\sigma|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in \downarrow_{coh}^{|\sigma|_\rho} a \right\} \\ |\lambda \alpha. \lambda x^\alpha. x^\alpha|_\rho &= \left\{ \langle a, (b, x) \rangle / b \subseteq \downarrow_{hom} a \text{ and } x \in \downarrow_{coh}^{\downarrow_{hom} a} (b) \right\} \\ |\lambda y^\sigma. \lambda x^\tau. x^\tau|_\rho &= \left\{ (a, (b, x)) / a \subseteq |\sigma|_\rho, b \subseteq |\tau|_\rho \text{ and } x \in \downarrow_{coh}^{|\tau|_\rho} (b) \right\} \end{aligned}$$

5.3 A candidate for β -completeness.

As announced in the preceding subsection, we drop the second part of the construction and take two trivial preorders. The model obtained is no longer extensional, but satisfies all the other properties of the BB -model stated so far, except for the domain equation on *types* : even if its web satisfies the same set-theoretical equation as for the BB -model, we now only have :

$$types \simeq (\cup_{l \in L} (P(N \cup \{l\}))) \oplus B \oplus \left(\bigcup_{D, D' \in types} P(D \Rightarrow D') \right)$$

where $(types \rightarrow types)$ is a retract of B ²⁵.

The interpretation of terms is much easier in it since we can now even drop all the internal \downarrow in the formulas above, and we get :

$$\begin{aligned} |\alpha|_\rho &\text{ is any homogeneous subset of } \Omega. \\ |\sigma \rightarrow \tau|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in |\tau|_\rho \right\} \end{aligned}$$

²⁵In fact $B = P^{hom}(\Omega_f^{hom} \times \Omega)$, is the set of all the homogeneous subsets of $\Omega_f^{hom} \times \Omega$, while $types \rightarrow types = S^{hom}(\Omega_f^{hom} \times \Omega)$ is only the set of those subsets which are furthermore \downarrow_{hom} -closed: even if we start from a trivial preorder on a web W , the induced preorder on $W \Rightarrow W$ will not be trivial ! Now, we are faced with two domains whose webs have the form $(\Omega_f^{hom} \times \Omega, \simeq_{hom}, \preceq)$ and $(\Omega_f^{hom} \times \Omega, \simeq_{hom}, =)$ respectively; in this case identity is trivially a morphism of webs and not an isomorphism, hence $types \rightarrow types$ is a proper retract of B , as stated.

$$|\forall\alpha.\sigma|_\rho = \left\{ \langle a, x \rangle / x \in |\sigma|_{\rho[\alpha:=a]} \right\}.$$

$|x^\sigma|_\rho$ is a coherent subset of $|\sigma|_\rho$.

$$\begin{aligned} |t^{\sigma \rightarrow \tau} u^\sigma|_\rho &= \left\{ x / \exists a \subseteq |u|_\rho \ (a, x) \in |t^{\sigma \rightarrow \tau}|_\rho \right\} \\ |\lambda x^\sigma . t^\tau|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in |t^\sigma|_{\rho[x^\sigma:=a]} \right\} \\ |t^{\forall\alpha.\sigma} \tau|_\rho &= \left\{ x / \exists a \subseteq |\tau|_\rho \text{ and } \langle a, x \rangle \in |t^{\forall\alpha.\sigma}|_\rho \right\} \\ |\lambda\alpha . t^\sigma|_\rho &= \left\{ \langle a, x \rangle / x \in |t|_{\rho[\alpha:=a]} \right\} \end{aligned}$$

Examples :

$$\begin{aligned} |\forall\alpha.\alpha|_\rho &= \left\{ \langle a, x \rangle / x \in a \right\} \\ |\forall\alpha.\alpha \rightarrow \alpha|_\rho &= \left\{ \langle a, (b, x) \rangle / b \subseteq a \text{ and } x \in a \right\} \\ |\lambda x^\sigma . x^\sigma|_\rho &= \left\{ (a, x) / a \subseteq |\sigma|_\rho \text{ and } x \in a \right\} \quad \dots \\ |\lambda\alpha . \lambda x^\alpha . x^\alpha|_\rho &= \left\{ \langle a, (b, x) \rangle / b \subseteq a \text{ and } x \in b \right\} \\ |\lambda y^\sigma . \lambda x^\tau . x^\tau|_\rho &= \left\{ (a, (b, x)) / a \subseteq |\sigma|_\rho, b \subseteq |\tau|_\rho \text{ and } x \in b \right\} \end{aligned}$$

Conjecture 11 *This model is complete for F .*

(Hint. The first thing to do would be to check whether the proof of [5] proves more generally that the theory of any polymax model which satisfies the completeness conditions is included in F_η .)

A still simpler model would be obtained by starting from the trivial homogeneity and coherence relations on A . But then we get all of $P(A)$ as the first component of *types*, and hence lose the evident way to construct adequate morphisms C , *index*, and *case* to fulfill the last “completeness conditions” of [5] (cf. section 7). The equational theory of this model would also be of interest.

5.4 A candidate for being an intermediate model.

By an *intermediate model* we mean a model \mathcal{M} such that $F \subsetneq Th(\mathcal{M}) \subsetneq F_\eta$, where $Th(\mathcal{M})$ denotes the equational theory of \mathcal{M} .

We simply do the same construction as for the *BB*-model, but start with an extra atom ω that we suppose all along to be coherent and homogeneous with everybody. We then get a model such that $X \Rightarrow Y \simeq (X \rightarrow Y)_\perp$, where Z_\perp denotes the “lifted” domain obtained by adding a new bottom element under the old one ; similarly, $Q(F) \simeq (types \rightarrow_F Terms)_\perp$ and the model is hence clearly nonextensional.

More precisely, it is easy to check that, at the level of webs, each element u of $X_{D \Rightarrow D'}$ is either the empty set or of the form $u = lmd_{D, D'}(f)$. In the first case $apl_{D, D'}(u)(v) = u = \emptyset$ for all v , and in the second case $apl_{D, D'}(u) = f$. From this observation and from the formulas defining the interpretation of terms in Section 2.3 we deduce that for all $t^{\sigma \rightarrow \tau}$ and for all x^σ not free in $t^{\sigma \rightarrow \tau}$, we have

$|\lambda x^\sigma . t^{\sigma \rightarrow \tau} x^\sigma|_\rho$ is either $|t^{\sigma \rightarrow \tau}|_\rho$ or $|\lambda x^\sigma . \emptyset|_\rho$. Similarly we have $|\lambda \alpha . t^{\forall \alpha . \sigma} \alpha|_\rho$ is either $|t^{\forall \alpha . \sigma}|_\rho$ or $|\lambda \alpha . \emptyset|_\rho$.

Conjecture 12 *This model is intermediate or all the models of the family are complete, either for F or for F_η .*

6 There is no stable analogue.

We explain here why there is no analogue of our class of models in the stable semantics. The following argument assumes that the reader is familiar with the stable semantics of simply typed calculus. In fact we can even assume, without loss of generality, that we are working with the *c.c.c.* of *coherent spaces* and *stable functions* (this background can be found in the first part of [16]). We just recall that a *coherent space* is a binary prime algebraic Scott domain whose web W is of the form (D, \approx) , where \approx is reflexive and symmetric (working with trivial preorders is more innocent in the stable semantics than in the continuous one). This *c.c.c.* will hence be our *Univ* here. The natural thing in the coherent semantics is to encode a stable function f between two spaces of web W, W' , by means of its “stable trace” $Tr_s(f)$, which is defined by:

$$Tr_{W, W'}^s(f) := \left\{ (a, x) \in D_f^{\approx} \times D' / a \text{ minimal such that } x \in f(a) \right\}$$

It follows easily from this definition that, if (a, x) and (b, y) are in some stable trace, then:

$$a \approx b \text{ and } x = y \text{ imply } a = b \text{ (*)}$$

Suppose now that we want to make a construction similar to that of the previous sections. Then, the web of a “stable webbed model of F ” would be of the form $(\Omega, \approx_{hom}, \approx_{coh}, j_{hom}, j_{coh})$, satisfying (only) four conditions, namely (5) and the variants of conditions (3,7, 8) where in the righthandside (*) is furthermore required (with respect to \approx_{hom} or \approx_{coh}).

types is now the set of homogeneous subsets D of Ω , ordered by inclusion, and each such D gives rise to the coherent space (D, \approx_{coh}^D) . Furthermore $Q, AP, Terms, \Rightarrow, apl, lbd, Lambda$ and $Appl$, can be defined, in such a way that for Q, lbd and $Lambda$ are defined as before, except that we replace Tr by Tr_s .

One can already note here that there would be no square models since (3,5) are obviously incompatible when $j_{hom} = j_{coh}$, but the fact that there is no model at all follows from remarks of a less technical kind. The first one is that $Lambda$ could no more induce morphisms $Lambda_F$ from $types \rightarrow_F Terms$ to $Q(F)$. In fact this would imply $Lambda(f) \subseteq Q(F)$ for all $f \in Hom_F(types, Terms)$, as in the continuous case; but here this implies that $Tr^s(f) \subseteq Tr^s(F)$ (since j_{hom} is injective), which is a drastic condition. In fact this is equivalent to $f \leq_s F$, where \leq_s is Berry’s order (see [16])²⁶. In particular it is easy to check that it forces f to be constant if F is constant. In other words our model, if any,

²⁶This comparison makes sense since $Terms \subseteq types$, hence f can also be viewed as belonging to $Hom(types, types)$.

would satisfy Axiom C (and hence would not be polymax). We would also be happy with this, since models of F_C are also interesting, and we could decide to define $Hom_F(types, Terms)$ by keeping only those $f \in Hom(types, Terms)$ which are such that $f \leq_s F (**)$. In this case $(Lambda, Appl)$ induces good retraction pairs $(Lamda_F, Appl_F)$, but unfortunately we do not have a model, because $(**)$ is not robust enough to allow us to prove Lemma 3 (the failure shows up when t^σ is $u^{\tau \rightarrow \sigma} v^\tau$).

7 Miscellaneous.

7.1 Testing the conditions for completeness.

The conditions for completeness presented in [5] divide into two groups. The first group is very easy to satisfy if $Univ$ is Scott's *c.c.c.* The second group consists of five conditions including : properties 5,6 in Section 2.2 (left-invertibility of Q and existence of a trace function), which are true in all our models, and the (quasi) left-invertibility of \Rightarrow , which will be true in many of them ; the two remaining conditions, which include the existence of a case function discriminating over types, happen to be rather easy to force in the present setting, and the BB -model happens to be the most natural model which satisfies them.

About the trace function. The trace "function" $j \in \forall D. \forall D'. (D \Rightarrow D')$ is defined on $types \times types$ by : $j(D)(D')(u) = \downarrow_{coh}^{D'} (u \cap D')$, and can then be raised to $Types \times Types$.

About the (quasi) left invertibility of \Rightarrow . Since E empty implies $D \Rightarrow E$ empty for all D , \Rightarrow cannot be really left invertible. So, we define : a (quasi) left inverse for \Rightarrow is a pair (P_1, P_2) of morphisms in $Hom(types, types)$, such that $P_1(D \Rightarrow E) = D$ if E is nonempty, and \emptyset otherwise, and $P_2(D \Rightarrow E) = E$ (always). Note that this could not happen in a stable semantics since P_1 cannot be stable.

But in our continuous class two further conditions are sufficient to ensure that \Rightarrow is (quasi) left invertible, namely: condition (12) below and the converse of (5). They are for example fulfilled if \approx_{hom} and \preceq_{hom} are trivial, that is to say when (W_{hom}, j_{hom}) is a graph model. The BB -model also fulfills them and other examples are given below.

The natural thing is to take :

$$\begin{aligned} P_1(D) &:= \downarrow_{hom} \{ y \in \Omega / \exists x j_{hcoh}(\{y\}, x) \in D \} \\ P_2(D) &:= \downarrow_{hom} \left\{ x \in \Omega / \exists a \in \Omega_f^{hcoh} j_{hcoh}(a, x) \in D \right\} \end{aligned}$$

Then $P_1(D)$ and $P_2(D)$ are homogeneous provided the converse of (5) holds, and it is then obvious that P_1 and P_2 are continuous.

$P_2(D \Rightarrow D') = D'$ is clear.

$P_1(D \Rightarrow D') = \emptyset$ if D' is empty is clear,

$P_1(D \Rightarrow D') \supseteq D$ is clear, if D' is nonempty

but the converse calls for some extra property, namely that, for all $a, b \in \Omega_f^{hcoh}$ and $x, y \in \Omega$,

$$j_{hcoh}(a, x) \preceq_{hom} j_{hcoh}(b, y) \implies (a, x) = (b, y) \quad (12)$$

(otherwise $P_1(\emptyset \Rightarrow D')$ may be nonempty, for example).

Example 13 *These two further conditions are satisfied by \mathcal{E}^2 , and more generally by all graph²-models, and also by the BB-model and by the candidate model of section 5.3. They are not satisfied by any of the (ext-K)²-models. In particular these conditions are independent from extensionality.*

About the existence of a case function on types. We mentioned in the introduction that the BB-model has a case function which is able to discriminate over types, and can in particular distinguish whether a nontrivial semantic type is some $Q(F)$ or some $X \Rightarrow Y$. Such a function does not exist in all our continuous models. For example it does not exist in models such that Q is onto; to obtain such a model it is enough to choose W_{hom} such that *types* is an extensional model of untyped λ -calculus.

7.2 Comparison with the *u.r.* models.

Univ is the category of Scott domains, or complete lattices in some cases, and continuous functions. First one fixes a reflexive object (M, q, ap) of *Univ*, namely a model of untyped λ -calculus in this category. Then, we let $c \in M$ be the code in M of a universal finitary retraction [24],[32], or a universal finitary projection [1], or a universal closure [31],[23]. In the stable semantics all retractions are finitary and c can be taken as the code of any ²⁷ universal (stable) retraction [4].

The necessary background and the missing proofs below can be found in [7], which surveys the preceding works.

For all *u.r.* models we take :

Terms := M .

types := $\{ cu / u \in M \}$, ordered by the restriction of the order on M , say \leq , is a Scott domain. The elements of *types* are hence all the retractions of M which belong to the class for which c is universal.

Types := $\{ rM / r \in types \}$. All r 's are finitary, hence each rM is a Scott domain, when ordered with the restriction of the order on M , and is a sub *p.o.* of *Terms*.

Thus “ r ” plays the role of “ D ” and “ rM ” that of X_D , in the definition of polymax models

Remark. When c is a universal finitary projection (*Types*, \subseteq) is isomorphic to (*types*, \leq). When c is a universal closure (*Types*, \supseteq) is isomorphic to (*types*, \leq).

²⁷Universal projections or closures are necessarily unique, but there is an infinite number of universal retractions, at least in the stable semantics (cf. [17]).

However, when c is a universal finitary retraction, $r \mapsto rM$ is not 1-1, and there is no way to put an order on $Types$ which would make $Types$ an homomorphic image of $types$. This justifies the choice of a more general variant of the definition of $polymax$ in the present paper.

Notations. In the following $q(f)$ is also denoted by $\lambda x.f(x)$, for $f \in Hom(M, M)$, and $ap(u)(v)$ is simply noted uv . Finally “ \circ ” denotes as well composition of functions or of their codes, following usual abbreviations in λ -calculus.

$$r \Rightarrow r' := \lambda z.r' \circ z \circ r, \text{ for } r, r' \in types.$$

$$Q(F) := \lambda z.\lambda x.(F(cx) \circ z \circ c)x, \text{ for } F \in Hom(types, types).$$

$$Lambda(f) := \lambda x.f(cx), \text{ for } f \in Hom(types, Terms).$$

$$Appl(u)(r) := ur \text{ (i.e. } Appl := ap).$$

$$lbd_{r,r'}(f) := \lambda x.r'(f(rx)), \text{ for } f \in Hom(rM, r'M).$$

$$apl_{r,r'}(u) := x \in rM \mapsto r'(ux), \text{ for } u \in Terms \text{ and } r \in types.$$

An amazing observation. All $u.r.$ models give rise to extensional models of F , even if we started from a nonextensional model of untyped λ -calculus. To support this claim we just have to check that for all r, r' we have $(lbd_{r,r'} \circ apl_{r,r'})(u) = u$ for all $u \in (r \Rightarrow r')M$ and that $(Lambda \circ Appl)(u) = u$ for all $u \in Q(F)$ and all F , which is easily done (using in particular that $u = \lambda x.ux$ for all retraction u).

In particular this shows that the webbed model \mathcal{E}^2 is definitely different from the $u.r.$ models of F built over \mathcal{E} (in the next paragraph we will compare the interpretations of a simple term in these models). Finally we suspect that Q is not left-invertible in $u.r.$ models of F , in contrast with webbed models.

Comparing the interpretations of $\lambda\alpha.\lambda x^\alpha.x^\alpha$ in the continuous $u.r.$ models based over Engeler’s model \mathcal{E} , with its interpretation in \mathcal{E}^2 , namely :

$$|\lambda\alpha.\lambda x^\alpha.x^\alpha| = \{ (a, (b, x)) / b \subseteq a \text{ and } x \in b \}.$$

The interpretation of this term in any $u.r.$ model based on \mathcal{E} (see [7, p.76]) is :

$|\lambda\alpha.\lambda x^\alpha.x^\alpha| = |\lambda x.\lambda y.cxy| = c$, where the second member is the interpretation in \mathcal{E} of a closed term of untyped λ -calculus with parameter c . From the equation on the left, and from the usual interpretation of untyped terms in \mathcal{E} , we deduce :

$$\begin{aligned} |\lambda\alpha.\lambda x^\alpha.x^\alpha| &= \{ (a, (b, x)) / x \in cab \} \\ &= (a, (b, x)) / \exists a' \subseteq a \exists b' \subseteq b (a', (b', x)) \in c \}. \end{aligned}$$

These two interpretations of $\lambda\alpha.\lambda x^\alpha.x^\alpha$ are necessarily different : otherwise $u \mapsto cau$ would not even be monotone, since for a, b finite one would have $cab = b$ if $b \subseteq a$ and $cab = \emptyset$ otherwise.

That our interpretation is simpler than the $u.r.$ one can easily be seen when interpreting more complex terms, or simply by comparing the interpretations of “ \rightarrow ”.

7.3 Are all the conditions defining F -webs necessary ?

Most of our eleven conditions are obviously necessary for building models in the line we did. There are only three conditions which were less natural, namely conditions (4), which requires that \preceq_{hom} should be included in \preceq_{coh} , and conditions (11,10). These three conditions were used for proving that *Appl* and *Lambda* could generate a family of retraction pairs allowing us to interpret abstractions over type-variables and application of a polymorphic term to a type (Proposition 9). Conditions (11,10) are also obviously linked to the fact that we encode Q and *Lambda* via their traces, while condition (4) was also useful for building *Terms*. We focus now on this last use of (4).

The operator \mathcal{S} presented in section 3 is well suited to build the domain *Types* as well as its elements, but it is only because we added condition (4), that we could use it for building *Terms*. This condition is of no direct need for modeling F . We could avoid it by building *Terms* via a variant of \mathcal{S} , called \mathcal{S}° below. However we did not push however the computations far enough to be able to claim that we would really have a model at the end, that is to say that we could really get completely rid of condition (4). It is already clear that suppressing it leads to technical difficulties.

For the reader interested in eliminating (4) we give below the definition of \mathcal{S}° , preceded by some preliminaries.

A *closure* is any continuous function cl on $P(D)$, D any set, such that $cl^2(a) = cl(a)$ and $cl(a) \supseteq a$ for all $a \subseteq D$. An example of a closure is the \downarrow defined above from a preorder \preceq on D , which we will consider as the *canonical closure* associated with \preceq . Suppose now that \preceq is the intersection of two preorders \preceq_1 and \preceq_2 , and let \downarrow_1 and \downarrow_2 be their canonical closures ; then, besides the canonical closure associated with \preceq there is another one, namely \Downarrow defined by : $\Downarrow a := \downarrow_1 a \cap \downarrow_2 a$. It is worth noting that \Downarrow -closed subsets of D are \downarrow -closed, and that \Downarrow and \downarrow do coincide on singletons, and that they coincide everywhere in the case where one of the preorders refines the other one.

Suppose now that $W := (D, \varphi, \preceq)$, where $\varphi := \varphi_1 \cap \varphi_2$ and $\preceq := \preceq_1 \cap \preceq_2$, and $W_1 := (D, \varphi_1, \preceq_1)$ and $W_2 := (D, \varphi_2, \preceq_2)$ are two prime webs. Then W is a prime web but from W we can now define two prime algebraic domains : namely $\mathcal{S}(W)$ and $\mathcal{S}^\circ(W)$, where $\mathcal{S}^\circ(W)$ is the set of \Downarrow -closed elements of $\mathcal{S}(W)$, ordered by inclusion.

$\mathcal{S}^\circ(W)$ and $\mathcal{S}(W)$ are very similar. We have that $\mathcal{S}^\circ(W) \subseteq \mathcal{S}(W)$ and the former domain is a retract of the second one (via \Downarrow) ; both domains admit the same prime elements, namely the sets of the form $\Downarrow \{x\} = \downarrow \{x\}$, and in both domains each element a is the union and the sup of the prime elements below it ; the only difference is that $\mathcal{S}(W)$ is closed under all unions, while $\mathcal{S}^\circ(W)$ is only closed under directed unions, hence sup is *not* union in $\mathcal{S}^\circ(W)$ (but *is* union for directed sets of subsets).

Finally, the two constructions coincide when \preceq_1 refines \preceq_2 .

Acknowledgements. Many thanks to the anonymous referee for his careful reading, his pertinent remarks, and for improving our English.

References

- [1] R. Amadio, K. Bruce, and G. Longo, The finitary Projection Model for 2nd Order Calculus, IEEE Conference on Logic in Computer Science, Boston, Juin 1986.
- [2] F. Barbanera, S. Berardi, A domain of domains model of polymorphism, technical report, University of Turin, 1997.
- [3] H. Barendregt, The lambda-calculus, its syntax and semantics, Studies in Logic vol. 103, North Holland, revised edition 1984.
- [4] S. Berardi, Retractions on dI-Domains as a Model for Type:Type, Information and Computation 94, p.204-231, 1991.
- [5] S. Berardi and C. Berline, A $\beta\eta$ -complete model for System F , preprint, Juin 1998, available on the authors' web pages.
- [6] S. Berardi and C. Berline, Easy models for $F_\omega Y$, Calculus of Construction, and *Types : Types*, in preparation.
- [7] C. Berline, Retractions et interprétation interne du polymorphisme : le problème de la rétraction universelle, Informatique Théorique et Applications/Theoretical Informatics and Applications, vol.26, n°1, p.59-91, 1992.
- [8] C. Berline, From Computations to Foundations : the λ -calculus and its webbed models, revised version 86 p., to appear in Theoretical Computere Science.
- [9] M. Coppo, M. Dezani-Ciangaglini, F. Honsell, and G. Longo. Extended type structures and filter λ -models, In G. Longo, G. Lolli and A. Marcja, editors, Logic Colloquium'82, p.241-262, Elsevier Science Publishers, 1984.
- [10] T. Coquand, C. Gunter, G. Winskel, DI-domains as a model of polymorphism, 3rd Workshop on Mathematical Foundations of Programming Language Semantics (April 87), LNCS 298, p.344-363, Springer-Verlag, 1987.
- [11] T. Coquand, C. Gunter, G. Winskel, Domain theoretical models of polymorphism, Information and Computation, vol.81, p.123-167, 1989.
- [12] E. Engeler, Algebras and combinators, Alg. Univ. 13 (3), p.289-371, 1981.
- [13] H. Friedman, Equality between functionals, In R. Parikh, editor, Logic Colloquium 1972-1973, Lecture notes in Mathematics 453, p.22-37, Springer-Verlag, New-York, 1975.

- [14] J.Y. Girard, Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures et à la théorie des types. Proceedings of the 2nd Scandinavian Logic Symposium, J.E. Fenstadt, ed., p.63-92, North Holland, 1971.
- [15] J.Y. Girard, Interprétation fonctionnelle et élimination des coupures de l'Arithmétique d'ordre supérieure, Thèse, Université Paris 7, 1972.
- [16] J.Y. Girard, System F of variable types, fifteen years later, Theoretical Computer Science 45, p.159-192, 1986.
- [17] Y. Jiang and X. Gouy, Universal retractions on dI -domains, Information and Computation, vol.119, n°2, p.252-257, 1995.
- [18] J.L. Krivine, Lambda-calcul, types et modèles, Masson, 1990.
- [19] J.L. Krivine, Lambda-calculus, types and models, Ellis-Horwood, 1993.
(augmented english translation of the above).
- [20] K.G. Larsen and G. Winskel, Using information systems to solve recursive domain equations, In Lecture Notes in Computer Science, vol. 173 (Semantics of data types), p. 109-130, Spriger-Verlag, 1984.
- [21] G. Longo, K. Milsted and S. Soloviev, The genericity theorem and the notion of parametricity in the polymorphic λ -calculus, Theoretical Computer Science 121, p.323-349, 1993.
- [22] G. Longo, Set-theoretical models of λ -calculus : theories, expansions and isomorphisms, Annals of Pure and Applied Logic 24, p.153-188, 1983.
- [23] N. McCracken, An Investigation of a Programming Language with a Polymorphic Type Structure, PhD Dissertation, 126 p., Syracuse University, School of Computer and Information Science, Syracuse, New York, June 1979.
- [24] N. McCracken, A Finitary Retract Model for the Polymorphic λ -calculus, unpublished, 1984.
- [25] M. Nielsen, G. Plotkin and G. Winskel, Petri Nets, event structures and domains, Part I, Theoretical Computer Science 13, 1981.
- [26] P. O'Hearn, Parametric Polymorphism, in: M. P. Fiore, A. Jung, E. Moggi, P. O'Hearn, J. Riecke, G. Rosolini and I. Stark, Domains and Denotational Semantics : History, Accomplishments and Open Problems, Bulletin of EATCS, vol.59, p.227-256, 1996.
- [27] D. Park, The Y combinator in Scott's Lambda-calculus models, Theory of Computation report 13, Department of Computer Science, University of Warwick, 1976.

- [28] G. Plotkin, A set-theoretical definition of application, Memorandum MIP-R-95, School of artificial intelligence, University of Edinburgh, 1972.
- [29] J. Reynolds, Toward a theory of type structures, Colloque sur la programmation, SLNCS 19, p.408-425, 1974.
- [30] J. Reynolds, Types, Abstraction and Parametric Polymorphism, in: Information Processing 83, Proceedings of the IFIP 9th World Computer Science Congress, Paris, September 1983, ed. by R.E.A. Mason, Elsevier Science Publishers B.V. (North Holland), Amsterdam, 1983, pp.513-523.
- [31] D. Scott, Data types as lattices, SIAM Journal of Computation, vol.5, n°3, p.522-587, 1976.
- [32] D. Scott, A space of retracts, manuscript, 1980
- [33] D. Scott, Domains for Denotational Semantics, in LNCS 140, Springer-Verlag, ICALP'82, p.577-613, 1982.
- [34] A. Simpson, Categorical completeness results for the simply-typed λ -calculus, in : M. Dezani-Ciancaglini and G. Plotkin eds., Lecture Notes in Computer Science vol. 902, TLCA'95, p.414-427, Springer Verlag, Berlin, 1995.
- [35] A.S. Troelstra, Notes on second order arithmetic, in : Mathias and Rogers, eds., Cambridge Summer School in mathematical Logic, Lecture Notes in Mathematics 337, p. 171-205, Springer Verlag 1973.

S. Berardi,

Dipartimento di Informatica,
 Università degli Studi di Torino,
 Corso Svizzera 185,
 10149 Torino, ITALIA.
e-mail: stefano@di.unito.it.
URL: <http://www.di.unito.it/~stefano>

C. Berline,

Equipe de Logique mathématique (case 7012),
 CNRS-Université Paris7,
 2 place Jussieu,
 75251 Paris cedex 05, FRANCE.
e-mail: berline@logique.jussieu.fr
URL: <http://www.logique.jussieu.fr/www.berline/index.html>