http://www.logique.jussieu.fr/www.berline/index.html
CNRS-Université Paris 7-France

Simple models for System F

MFPS [*]
May 10-May 13, 1998
Queen Mary/Westfield College
University of London

1 Introduction.

The aim of this talk is to present two families of models for Girard and Reynolds System F, also called *polymorphic λ-calculus* (71,74)). These families relate respectively to Scott's continuous semantics and to Berry's stable semantics. The continuous family has been built, and its stable analogue is in progress.

System F, which plays a major role in theoretic computer science, happens to be difficult to model, because of its impredicative features, a few models have been proposed, as well as a few abstract definitions of models of F. None is completely satisfactory so far.

Our proposal is not completely satisfactory either[1], but it is is easy to understand, allows to build and control directly a great variety of models and, hopefully, to study in depth the ones were the interpretation of terms is the most simple.

In particular **no functorial notions** is required, and, if one wants to use some category terminology to put things in context, as I will do here, then knowing what is a cartesian closed category (ccc) is enough.

These models **belong to a classical denotational setting (the continuous and stable semantics)**, but differ from the ones studied so far[2], namely[3]:

[*]This is in fact a slightly expanded but still informal version of the talk (updated May 22th, slightly revised Sept 28th). Some further properties of our models are explicited, which should help for comparison with other models, but which were out of the scope of a talk for a general audience.

[1]Because our models are highly non parametric.

[2]At least the ones I know.

[3]Dana Scott pointed out at the end of the talk that realizability models also belong to this setting, which I missed out.

U.R. Models (Scott 75-80, McCracken 84, Bruce-Amadio-Longo 86, Berardi 91). Here "$u.r.$" stands for "universal" retraction.

Categorical models (Girard 86 and Coquant-Gunter-Winskel 89 (CGW for short)).

These models are however **not revolutionary**, in particular they share enough features with these models so that comparison is possible, and in a deeper way that I will do here. In fact no ingredient is new, but combining them that way seems not to have been considered before.

In the course of the talk I will **propose a very simple-minded notion of "models of F"**, which works for 4 kinds of models : The trivial ones, the continuous or stable $u.r.$ models, and, and the continuous class we present here. It is too simple to cover the other models, and in particular our stable class, but helps for understanding.

The models in our **continuous** class are **rich in polymorphic maps**, which is maybe drawback for some questions, but is certainly a blessing for other ones. In particular this allows us to present a **complete model** for $F\eta$, whose definition has nothing to do with syntax, and to understand why it works. **The stable class** happens to be naturally more constrained.

Many of our models here can be seen as two **intertwined models of untyped λ-calculus**, but the classes are richer than that. All the models could on the other hand be described as two intertwined **intersection type systems**[4], with two implications and two restricted versions of the set-theoretic conjunction.[5]

How did these models arise.

S. Berardi wanted to exhibit a complete model for $F\eta$ ($F\eta C$ at the beginning).

For this purpose he built **with F.Barbanera** a (continuous) model, called here the **BB-model** for short, viewing it as a conceptual simplification of the Girard and CGW model and conjectured that this model was complete for $F\eta$ (and in fact it is [Berardi-Berline, preprint end of May 98]), and that its stable analogue, if any existed, could be complete for $F\eta C$, the extension of F studied by Longo-Milsted-Soloviev).

My choice was to **view the BB-model as one occurrence of a general family**, related to classical models of untyped λ-calculus. The interest of this view is, first that it helps to understand the BB-model, and second that the class happens to contain very simple models (especially simple when non extensional).

[4]This alludes to type assignments systems "à la Dezani & als.". The use of restricted systems where (when defining types) conjunction and the empty set are not allowed on the righthandside of an implication, is highly recommended. In this case, this view is transparently equivalent to the definition of the a model via an algebraic "web" (in the sense of Section **??**).

In fact, to meet the implicit requirements of a typing system one should restrict to recursively presented webs, which is maybe more natural in a computer science setting. But this would rule out many models for unnatural reasons, from a mathematical point of view.

[5]Of course for the simplest models the two implications and/or conjunctions may coincide.

Then I began to study the **stable analogues,** which all happen to be models of FC, the extension of F put in the light by Longo-Milsted-Soloviev 93.

This is work in progress[6], which highly benefits of the frequent exchanges we have with S. Berardi about the completeness problem.

2 System F and extensions.

System F is the fragment of λ-calculus which can be controlled by 2nd order propositional calculus (over $\to$ and $\forall$). It dates back to Girard 71 and Reynolds 74. Each F-term $t : \sigma$ encodes a proof of σ in this logic.

Simply typed λ-calculus is its restriction to $\to$ and is much older.

Interest of F in theoretic computer science :

(1) can **encode a wide class of data types** (booleans, integers, lists-of-something, trees-of-something, aso.) (at the logical level) (ex: $N := \forall\xi.((\xi \to \xi), \xi \to \xi)$).

(2) can **encode all ever interesting computable total functions** on such data types.

(3) **allows uniform programming (=polymorphism)**, at the level of types.

Ex: there is an abstract type $List$, and not only types $List(\sigma)$ for some specific σ's),and of programs : there exists $t : \forall\xi.(List(\xi) \to List(\xi))$ which reverses all possible finite lists.

(4) Because of its expressive power, F **gives wide interest to the Curry-Howard** Correspondence between proofs and programs (written as F-terms), all the more that F is the core of still more expressive systems, to begin with *System* $AF2$ of Leivant and Krivine, which is based on second order predicate logic and ensures a very strong correction of programs, since one can specify exactly in it any function one may want to compute.

- **F-types :**

  = second order propositional formulas built from (type) variables, $\to$ and $\forall$.

$$\sigma := \ \xi \mid \sigma \to \sigma \mid \forall\xi.\sigma$$

- **F-terms** : Restrictions on the three last formation rules.

[6] In particular the claims that are made here about the stable models are at the moment only supported by disconnected proofs : no systematic presentation has been yet written down. The continuous case is, on the other hand, is not far from being completely written down, and will give rise to a join paper with S. Berardi, entitled "Building continuous models for System F".

Each term comes with a fixed implicit type, that is made explicit below :

$$\begin{array}{rcl} x^\sigma & : & \sigma \\ \lambda x^\sigma.t^\tau & : & \sigma \to \tau \\ t^{\sigma\to\tau}u^\sigma & : & \tau \\ \lambda\xi.t^\tau & : & \forall\xi.\tau \text{ provided } \xi \text{ ...} \\ t^{\forall\xi.\tau}\sigma & : & \tau[\xi : \sigma] \end{array}$$

- $\forall$ and the two $\lambda's$ are binders. The condition restricting ξ, two lines above, is that it should not be free in a free term variable of t. On the last line $\tau[\xi : \sigma]$ means that σ replaces all free occurrences of ξ in τ, after a possible renaming of bounded variables of τ to avoid variable clashes, and similar notations are used for terms below.
- By a **polymorphic type or term** we will understand here those types and terms which begin with $\forall\xi$ or $\lambda\xi$.

  Ex: The polymorphic identity $\lambda\xi.\lambda x^\xi.\mathbf{x}^\xi : \forall\xi.\xi \to \xi$.
- **Immediate** β-reduction is defined by :

$(\lambda x^\sigma.t)u^\sigma \to t[x^\sigma : u^\sigma]$
$(\lambda\xi.t)\sigma \to t[\xi : \sigma]$.
Immediate η**-reduction** is defined by :
$\lambda x^\sigma.tx^\sigma \to t$ if $t : \sigma \to \tau$ for some τ and x^σ is not free in t.
$\lambda\alpha.t\alpha \to t$ if $t : \forall\alpha'.\sigma$ for some α', σ and α is not free in t.
Immediate $\beta\eta$**-reduction** as the union of both.

Then we define β**- (resp.** $\beta\eta$**-) reduction** as the smallest transitive relation containing the corresponding immediate reduction, and compatible with the formation of terms. Reduction preserves typing. Finally we define β**- (resp.** $\beta\eta$**-) equivalence**. as the reflexive, symmetric and transitive closure of the corresponding reduction ; these relations are denoted by $=_\beta$ and $=_{\beta\eta}$.

2.1 Consistent extensions of F.

In this talk we will only be interested with the equivalence relations and

F **will denote** β**-equivalence, while** $F\eta$ **will denote** $\beta\eta$**-equivalence.**

FC **(resp.** $F\eta C$) is then the extension of F (resp. $F\eta$) by the set of the equations of the form $t^{\forall\xi.\sigma}\tau = t^{\forall\xi.\sigma}\tau'$ for all $t, \sigma, \xi, \tau, \tau'$ such that ξ is not free in σ. Stated informally, this set of axioms, called "Axiom C" expresses that constant polymorphic types can only contain constant polymorphic maps.

Finally we mention that it is consistent to add to F (or $F\eta$) the fact that there is a type of types. The corresponding extension of F, which is denoted by $Types : Types$, will not be formally described here.

3 The continuous and stable semantics.

We now want to **give a meaning to types and terms** in a mathematical structure, in such a way that two equivalent terms get the same interpretation. Interpretations are done relatively to environments ρ, which give values to free variables. We add obvious compatibility requirements like, $|t^\sigma|_\rho \in |\sigma|_\rho$ if $|\sigma|_\rho$ is a set.

When η-equivalent terms are equated, then one speak of **extensional models.**

In this subsection we stick to simply typed λ-calculus.

One interpret types as objects of a category $TYPES$, which has to be a ccc[7] in order the interpretation fulfills the preceding requirements.

(Here I will only consider categories where objects are **true sets** and morphisms **true functions**).

We define $|\sigma \to \tau|_\rho$ as $Hom(|\sigma|_\rho, |\tau|_\rho)$, the full set of morphisms between both.

The main classical semantics for simply-typed λ-calculus are :

The set-theoretical semantics, were $TYPES := ENS$.

Scott's continuous semantics, were $TYPES$ is some ccc of Scott domains (partially ordered sets with directed sups, a.s.o.) and the morphisms are the Scott continuous functions between them (monotone and commuting with directed sups).

Berry's stable semantics, were $TYPES$ is some ccc of Scott domains (which admits relevant infs) and the morphisms are the stable functions, namely those continuous functions which commute with the relevant infs.

For the two last semantics **several choices of classes of domains** are possible, but the sets of morphisms is fixed. $TYPES$ can be small (a set), closed under some avatar $\to$ of Hom.

4 Extension to F is not obvious.

This section refers to the intuitive and **informal notion of model above**. We will make things precise later on.

First : there is **no set-theoretic semantics of** F (Reynolds 84).

Fortunately, **extending the other ones is possible** !

However the solutions used so far are very involved.

The problem is of course due to $\forall\xi.\sigma[\xi]$.

Let us view σ as a "constructor over types" (which associates $\sigma[\xi : \tau]$ to any type τ).

It is natural, in the situation above to interpret σ by a "constructor" $F : TYPES \to TYPES$, and any $t : \forall\xi.\sigma[\xi]$ as a "function" $f \in \Pi_{X \in TYPES} F(X)$,

[7] more precisely : a ccc with "enough points" ; this is a technical condition which will always be understood here (see e.g. Barendregt's book for details).

that is $f : TYPES \to \cup TYPES$, such that $f(X) \in F(X)$ for all X in $TYPES$. **The first difficulty** is to give a meaning to "f : ...", **the second** to get a set (a domain) for $|\forall\xi.\sigma[\xi]|_\rho$, **the third** is the nature of F (functor ?), of f ?, and **last but not least**, to have everything work together (so that one could define inductively the interpretation of types and terms).

Thus one is lead to make drastic restrictions on $TYPES$, or/and on the possible F and f. First to be able to give an accurate definition, and second to decrease the level of concepts (optimal reduction : *Types* to an object, *functors* to morphisms).

Let us call informally **polymorphic maps related to** F the $f \in \Pi_{X \in TYPES} F(X)$ which will be retained at the end (in the considered model), it happens that the same interpretation of $\to$ is compatible with **different choices of classes** of polymorphic maps[8].

We have two examples for the continuous semantics : no restriction for *u.r.* models and for our continuous models, strong restriction for the continuous CGW).

And we have three for the stable semantics : no restriction for *u.r.* models, mild restriction for our stable models, and finally drastic restriction for Girard's model.

I would say that at present there is **no real understanding of the possible links between the interpretation of $\to$ and that of $\forall$.** There are general categorical pictures, (in Asperti-Longo's book for example), they are however very conceptual and for me completely opaque. An intermediate picture using some category but with no pretention to cover all models would probably more enlightening, but this is still to be done. In fact people like Girard or CGW built models, proved that they model F, but did not try to say what was a model of F.

Remark. If the model is such that for all constant functor F all polymorphic maps are constants, then the model is a model of FC, and hence cannot be complete *w.r.t.* F or $F\eta$!

This is the case of a lot of the models below.

5 Previous models.

- Realizability and PER models (Troelstra, Girard 71, ...).[9]

[8] Each giving rise to infinitely possible choices for the interpretation of $\forall$ (one for each non extensional model, roughly saying)..

[9] As pointed out by Dana Scott the class of realizability models is indeed very rich and is related to the continuous or stable semantics. He suggested that there could be deep relations with our models. At the present moment (May 22th) I did not yet found the time to look at them, and to begin with, I have not checked whether the realisability models all satisfy FC, like all PER models, which I implicitely thought, or not. A positive answer to this question, implies that our continuous models would definitely be different from the realizability ones, since none of them satisfies FC. For the stable case another argument should be given, using maybe that our models encode morphisms via traces, while realizability models would hopefully encode morphisms directly. This last point is not yet clear for me.

Good for proving programming properties, **not considered here**.

Has only constant polymorphic maps, hence are models of FC.

- **Trivial models** One fixes a model of untyped λ-calculus and interpret F by forgetting all typing indications.

  This corresponds to a **very drastic restriction on** $TYPES$, which is here a singleton !, in particular all polymorphic maps are constant !

  Plenty of models in each ccc, all are models of FC.

- **U.R. models** (Scott 75-80, McCracken 84, Bruce-Amadio-Longo 86, Berardi 91).

  Once more one fixes a model of untyped λ-calculus M, and furthermore a "universal" retraction of this model (three kinds of models, according to the family of retractions that we consider[10]). One encodes F-types and F-terms as elements of M, using r as a parameter. Here, $TYPES$ is a set (in fact a domain of domains).

  These models also model $Type : Type$.

  Plenty of models in each ccc. Equational theories ??

- **Categorical models.** The pioneering work is **Girard 86** (stable semantics on qualitative domains), adapted by **CGW** to the stable semantics on DI-domains (87), then to the continuous semantics (89).

  No restriction on $TYPES$, hence only one model for each ccc, hence a few for each semantics, but drastic restrictions on functors and polymorphic maps.

  Very drastic in the case of Girard's model, which is in particular a model of $F\eta C$.

  Less drastic in the case of the continuous CGW, which does not satisfy Axiom C.

  Examples of economy in polymorphic maps.

  $\forall\xi.\xi$ gives here the trivial (= one point) domain in both Girard and CGW. The difference appears at the level of $\forall\xi.\xi \to \xi$, which has 2 elements in Girard and is infinite in the continuous CGW.

  Note : here $\forall$ quantifies officially over all domains, however it is a consequence of these works that there exists an equivalent model were $TYPES$ is a domain of domains (hence a set), however functorial considerations **could not be avoided** and the construction is conceptually difficult to understand.

[10] In the continuous semantics 3 classes of retractions are known to have a universal element : finitary retractions, finitary projections, and finally closures.
In the stable semantics the class of closures is trivial (reduces to identity), and all retractions are finitary. Hence there are only two classes to consider, that of projections and the full class of retractions.
A survey on universal retractions can be found on my web page (its reference in fact).

- **The present classes.**

  $TYPES$ is a set, no functorial considerations, plenty of models in each classical ccc,

  maximum number of polymorphic maps in the continuous case. In the stable case we have automatically a restriction on polymorphic maps which is intermediate between Girard's stringent condition and complete freedom[11], and all our stable models are models of FC.

6 Models of untyped λ-calculus.

6.1 Untyped λ-calculus.

- **Terms :**

$$t := x \mid \lambda x.t \mid t\, t$$

- **Immediate reductions** are defined by :

$(\lambda x.t)u \to t[x : u]$ (βreduction)
$\lambda x.tx \to t$ if x is not free in t (η-reduction)
Then β- and $\beta\eta$**- reductions and equivalences** are defined as above.

6.2 Reflexive models of the ccc's above.

A retraction pair in a category is a pair $(f, g) : A \lhd B$ is a pair of morphisms such that $f : A \to B$ and $g : B \to A$ and $g \circ f = id$.

A model of untyped λ-calculus is a **reflexive object of a ccc**. That is, a triple (M, A, L) such that :

$$(A, L) : Hom(M, M) \lhd M$$

Then, for any term t and *any* environment $\rho : Vterms \to M$, one can defines $|t|_\rho$ by induction on t :
$|t\, t'|_\rho := A(|t|_\rho)(|t'|_\rho)$
$|\lambda x.t|_\rho := L(u \in M \mapsto |t|_{\rho[x:=u]})$

This is well defined exactly because of the ambient category is a ccc, and is compatible with β-equivalence because (A, L) is a retraction pair. It is compatible with $\beta\eta$-equivalence exactly if (A, L) is a pair of isomorphisms.

[11] In a stable analogue setting a map f can indeed only be a polymorphic map related to F iff $f \leq_s F$, where $\leq_s$ is Berry's stable ordering.

6.3 The Engeler-Plotkin model .

The simplest model of λ-calculus is **the Engeler-Plotkin model**, defined as follows.

We start from a set A of "atoms", we define inductively a set Ω as the smallest set such that

$$\Omega = A \cup (\Omega^f \times \Omega)$$

We take $M := (P(\Omega), \subseteq)$, and finally define :

$L(f) := \{ (a, \alpha) \in \Omega^f \times \Omega \ / \ \alpha \in f(a)\}$

$A(u)(v) := \{ \alpha \ / \ \exists a \ (a, \alpha) \in u \}$

It is immediate to check that this is a retraction pair of the ccc of say, complete lattices and continuous functions. Interpretation of terms is very simple in it, ex :

$|\lambda x..x| := \{ (a, \alpha) \ / \ \alpha \in a \}$

$|\lambda x \lambda y.xy| := \{ (a, (b, \alpha)) \ / \ \exists b' \subseteq b \ (b', \alpha) \in a \}$

Non extensional model of the continuous semantics.

Gives rise to 5 models of System F (3 continuous $u.r.$ models of system F, according that we parametrize with a universal cloture, or finitary projection, or finitary retraction, the trivial model, **and**, last, the model that we will describe later on).

6.4 Webbed models for untyped λ-calculus.

In fact a lot of models of the continuous semantics can be presented in a similar way, I call them webbed models of λ-calculus, or "reflexive prime webs". They are nothing else than the reflexive prime algebraic Scott domains, provided those models are presented from their "web" that is the structured set of their set of primes (simplifies Scott's info. systems, also related to event structures). As it is often stressed, at the level of domains such view allows to **get rid of inverse limit constructions**, but as it is much less often stressed : **it makes interpretations of λ-terms in models much more workable**, and **moreover allows you to build many models of λ-calculus, that you would never rich with inverse limit constructions.**

I begin with the simplest of them.

6.4.1 Graph models (Plotkin, Scott, Engeler).

Just a pair (Ω, j) where Ω is a set and j an injection : $\Omega^f \times \Omega \to \Omega$.

Take $M := (P(\Omega), \subseteq)$, take A, L as above except that (a, α) is, replaced by $j(a, \alpha)$.

Interpretation as easy as before.

6.4.2 pcs-models.

Cover all classical models of untyped λ-calculus and a lot of other ones.

By definition **pcs-models** are the models of λ-calculus associated to **reflexive prime webs** :

A prime web is a triple $W := (\Omega, \stackrel{\frown}{\smile}, \leq)$ such that $\stackrel{\frown}{\smile}$ is a **coherence relation** on Ω (i.e. reflexive and symmetric binary relation) and $\leq$ is a **preorder**; both **compatible,** in the sense below. First for $u, v \subseteq \Omega$ we define : u is coherent if its elements are pairwise coherent, $u \stackrel{\frown}{\smile} v$ if $u \cup v$ is coherent, $\downarrow u$ is the downward closure of u relative to the preorder, and finally $u \leq v$ means $\downarrow u \subseteq \downarrow v$. Then the compatibility condition is :

$$a \text{ coherent } \Longrightarrow \downarrow a \text{ coherent}$$

Define here Ω^f as the set of finite *coherent* subsets of Ω.

A reflexive prime web is a pair (W, j) such that j is an injection $j : \Omega^f \times \Omega \to \Omega$ compatible with $\stackrel{\frown}{\smile}, \leq$ in the following sense :

$$\begin{array}{rcl} j(a,\alpha) & \leq & j(b,\beta) \Longrightarrow b \leq a \text{ and } \alpha \leq \beta \\ j(a,\alpha) & \stackrel{\frown}{\smile} & j(b,\beta) \Longleftrightarrow (b \stackrel{\frown}{\smile} a \Longrightarrow \alpha \stackrel{\frown}{\smile} \beta) \end{array}$$

Consider $M := (S(\Omega), \subseteq)$, where $S(\Omega)$ **is the set of downward-closed coherent subsets of** Ω. Then M is a (prime algebraic) domain, and a retraction pair (L, A) can be defined in a very similar way as for graph models, **just restricting** the a's to be coherent, and replacing sets by their downward closure every time it is necessary.

Such a triple (M, L, A) is called a **pcs-model**.

pcs-models enter on the scene as soon as one wants that M satisfies some domain equation. For example $\leq$ is necessary if one looks for extensional models[12].

7 A simple minded way to interpret F.

Interest. The following abstract definition is too coarse to be very general, it however

1. helps to understand how we intend to model F,
2. covers our continuous class, as well as r.u. models[13] and trivial models,
3. allows to show that the sufficient conditions we give for being complete for $F\eta$, hold in fact for a rather wide range of models.
4. Would not hold for the stable analogue of our class.

The general features are simple :

Types is a set (with additional structure : think of a domain of domains).

F-types are interpreted as elements of *Types*,

F-terms will be interpreted in a set called *Terms*,

There are no functors, only morphisms,

[12]This is true in the continuous semantics, not in the stable one.

[13]at least those based on projections and closures.

One takes retraction pairs as often as one has to model an (abstraction, application) pair between terms.

By **a model for system** F we will mean the following tuple, which lives in an ambient ccc $Univ$ (think of a Scott ccc). We assume that the objects of $Univ$ are non empty sets (with additional structure) and that morphisms are true functions. We assume furthermore that for all objects A, B, $Hom(A, B)$ is also an object of $Univ$, also denoted $A \to B$.

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

where $Types$ and $Terms$ are two objects of $Univ$, $Q, \Rightarrow, Lambda$ and $Appl$ are four morphisms, and finally lbd and apl are two families of morphisms indexed by $Types \times Types$.

1. **We require** that $Types$ is non empty and that its elements are also objects of $Univ$. They are denoted by X, Y, a.s.o. Elements of $Types$ will interpret types, while terms will be interpreted in $\cup Types$.

2. **We require** that each $X \in Types$ is a substructure of $Terms$ [14].

3. $\Rightarrow \in Hom(Types \times Types, Types)$ **is a morphism** and it will be the interpretation of the arrow constructor on types.

4. **We require** that $(lbd_{X,Y}, apl_{X,Y})$ **is a retraction pair** : $(X \to Y) \lhd (X \Rightarrow Y)$

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

5. $Q \in Hom((Types \to Types), Types)$, will be the interpretation of second order quantification and $Q(F)$ will also be denoted $\forall X.F(X)$ for $F \in Types \to Types$.

6. **We require** that $(Lambda, Appl)$ is a retraction pair: $Types \to Terms \lhd Terms$.

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

7. **We require** finally that, for all $F \in Types \to Types$, and $f \in Types \to Terms$, if $f(X) \in F(X)$ for all $X \in Types$, then $Lambda(f)$ belongs to $Q(F)$, and conversely that for all $u \in Q(F)$ and all X, we have $Appl(u)(X) \in F(X)$.

[14] We are grateful to G. Roselini to have pointed out that a previous simpler formulation was unlikely.

Out of any 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 $[.]_\rho$ of F-types and F-terms such that $|t^\sigma|_\rho \in |\sigma|_\rho$, and which identifies β-equivalent terms ($\beta\eta$ if the retraction pairs are inverse isomorphisms (but this si not completely true for $(Lambda, Appl)$.

Remark 1 *Types is not a type.*

Remark 2 *$X \Rightarrow Y$ should not be confused with $X \rightarrow Y$; in the best case (extensionality) they are only isomorphic. Let us give concrete examples : in the BB-model, $X \Rightarrow Y$ is the cpo of traces of elements of $Hom(X,Y)$. In general it is a domain which is bigger than the cpo generated by the traces of morphisms. And similarly with $Q(F)$. In u.r. models $X \Rightarrow Y$ is the domain of a retraction, and is isomorphic to $X \rightarrow Y$. In trivial models M, $(M \Rightarrow M) = M$, and hence isomorphic to $M \rightarrow M$ iff the model is extensional.*

Remark 3 *The two last conditions, which imply in particular that $Hom_F(Types, Terms)$ is "included" in $Q(F)$ is very drastic and rules out the (bounded) versions of Girard and CGW, as well as our class of stable models, which are poor in polymorphic maps.*

8 The bi-Engeler model.

Belongs to the continuous semantics.

$Terms := M := (P(\Omega), \subseteq)$,

$Types : (\{\, P(D) \;/\; D \subseteq \Omega \,\}, \sqsubseteq)$, where $\sqsubseteq$ is such that

$Types$ is isomorphic to the auxiliary domain $types$ [15].

$types := (P(\Omega), \subseteq)$

Remark : Here $Terms$ is a model of untyped λ-calculus, $types = Terms$ and $\sqsubseteq$ equals $\subseteq$, but none of these three properties is true in general for the models of our continuous class.

The model is defined as follows, from $types$, and then the morphisms can easily be transferred to $Types$ using the isomorphism between $types$ and $Types$.

Here (a, α) is always supposed to be in $\Omega^f \times \Omega$, and Ω^f is the set of all finite subsets of Ω.

$D \Rightarrow D' :=$	$\{\, (a,\alpha) \;/\; a \subseteq D$ and $\alpha \in D'\}$	
$Q(F) :=$	$\{\, (a,\alpha) \;/\;\; \alpha \in F(a)\}$	for $F \in types \rightarrow types$
$apl_{D,D'}(u)(v) :=$	$\{\, \alpha \;/\; \exists a \subseteq v \;\; (a,\alpha) \in u \,\}$	for $u \in P(D \Rightarrow D')$, $v \in P(D)$
$lbd_{D,D'}(f) :=$	$\{\, (a,\alpha) \;/\; a \subseteq D$ and $\alpha \in f(a)\}$	for $f \in P(D) \rightarrow P(D')$
$Appl(u)(D) :=$	$\{\, \alpha \;/\; \exists a \subseteq D$ and $(a,\alpha) \in u\}$	for $u \in P(\Omega)$
$Lambda(f) :=$	$\{\, (a,\alpha) \;/\;\; \alpha \in f(a)\}$	for $f \in P(\Omega) \rightarrow P(\Omega)$

Examples :

$|\forall\xi.\xi| = \{\, (a,\alpha) \;/\;\; \alpha \in a\}$

[15] Thus, in this very particular case, $\sqsubseteq$ happens to be nothing else than $\subseteq$!

$|\forall\xi.\xi \to \xi| = \{\ (a,(b,\alpha))\ /\ \ b \subseteq a \ and\ \alpha \in a\}$
$|\lambda x^\sigma.x^\sigma|_\rho = \left\{\ (a,\alpha)\ /\ a \subseteq |\sigma|_\rho \text{ and } \alpha \in a\ \right\}$
$\left|\lambda\xi.\lambda x^\xi.x^\xi\right| = \{\ (a,(b,\alpha))\ /\ \ b \subseteq a \ and\ \alpha \in b\ \}$
$\left|\lambda y^\varsigma.\lambda x^\xi.x^\xi\right|_\rho = \{\ (a,(b,\alpha))\ /\ \ a \subseteq \rho(\varsigma),\ b \subseteq \rho(\xi) \ and\ \alpha \in b\ \}$.

Remark : Within the $u.r.$ models, with universal relation "r", the polymorphic identity would be interpreted by :.
$\left|\lambda\xi.\lambda x^\xi.x^\xi\right| = \{\ (a,(b,\alpha))\ /\ \ \alpha \in rab\ \}$
$= \{\ (a,(b,\alpha)\ /\ \exists a' \subseteq a\ \exists b' \subseteq b\ (a',(b',\alpha)) \in r\}$
where "r" codes the universal retraction.

This is more complicate than the interpretation above, and necessarily different (if both interpretations were equal, then $x, y \mapsto rxy$ would not even be monotone, since one would have $rab = b$ if $b \subseteq a$ and $rab = \varnothing$ otherwise).

Another indication that our models differ from the $u.r.$ ones. is that they have no reason at all for being models of $Types : Types$, while the $u.r.$models are.

9 Freedom.

1. There is no reason why one should stick to Engeler model, all graph models and in fact most pcs-model M will give an F-model that way, that we will call an "M^2-model", or a "bi-model" (better : a square-model !). But note : **there cannot exist square-models in the stable semantics !**

2. There is no reason why we should interpret the three abstractions by so similar set theoretic constructor. One could as well start with two notions of pairs, or equivalently, with two 1-1 maps $j, j' : \Omega^f \times \Omega \to O$.

3. There is no reason why we should take the same coherences and preorders for delimiting types and terms. Ex. the BB-model, a.s.o.

BUT we can't of course intertwine pcs-models in a totally anarchic way. There is a global set of **11 compatibility conditions** which make things work. They are analogous to the 3 ones we gave for the pcs models of untyped λ-calculus. They are **easy** to check, to disprove, and, better**, to fulfill in many various ways,** especially when some of the relations are trivial **!**). They become very natural if one understands them in terms of morphisms of webs.

4. We do not need in fact to intertwine two models of untyped λ-calculus : this corresponds to F-models in which $Types$ as well as $Terms$ are themselves models of untyped λ-calculus, which is far stronger than necessary. However, all our models will be such that $Types$ is a model of untyped λ-calculus, and this is forced by the theoretical specificities of our modeling ; as noted earlier $Terms$ need not be, even in our setting.

THIS has analogues in the stable semantics[16].

[16] At least, there are good reasons to think so (at the moment of the conference).

10 Webbed continuous models for System F.

Bi-prime-webs. These are 5-tuples

$$W := (\Omega, \asymp_{\mathrm{hom}}, \asymp_{coh}, \leq_{hom}, \leq_{coh})$$

such that :

1. $\asymp_{\mathrm{hom}}$ and $\leq_{hom}$ (resp. $\asymp_{coh}$ and $\leq_{coh}$are compatible),

2. $\leq_{hom} \subseteq \leq_{coh}$ (really necessary ? at least very convenient).

Intuitively : $u \subseteq \Omega$ is homogeneous if it is a subset of some (non unique) type,

$u \subseteq \Omega$ is homogeneous and coherent if it is a subset of some element of a type.

We let Ω^f_{hom} and Ω^f_{hcoh} be the sets of finite homogeneous (resp. homogeneous and coherent) subsets of Ω.

Continuous webbed models for System F :

$$(W, j_{hom}, j_{hcoh})$$

with $j_{hom} : \Omega^f_{hom} \times \Omega \to \Omega$ injection

and $j_{hcoh} : \Omega^f_{hcoh} \times \Omega \to \Omega$ injection

Both compatible with W above (adds 8-9 conditions).

Then define :

types $:= (S(\Omega, \asymp_{hom}, \leq_{hom}), \subseteq)$ and, for $D \in$ *types*,

$X_D := S(D, \asymp^D_{coh}, \leq^D_{coh})$, where $\asymp^D_{coh}$ and $\leq^D_{coh}$are the restrictions of $\asymp_{coh}$ and $\leq_{coh}$ to D.

$Types := (\{\ X_D\ /\ D \in \textit{types}), \sqsubseteq)$ isomorphic to *types*.

$Terms := S(\Omega, (\asymp_{hom} \cap \asymp_{coh}), \leq_{hom}) = \cup Types$.

a.s.o.

11 Interest of these models.

1. They are **understandable,**

2. They are **feasible**, and one can hope that their equational theory is reachable, and that they may help to test the validity of hypothesis on F (checking whether one extension is consistent). All the more that :

3. They have **nice supplementary properties**, the simplest being that :

a. $Terms = \cup Types$

b. Q is left invertible (hence $Types$ is a model of untyped λ-calculus).

c. $\Rightarrow$ is left invertible in a lot of them,

d. Have a polymorphic "trace function" $h \in \forall X.\forall Y.X \Rightarrow Y$ such that $h(X,Y)(x) = x$. for all $x \in X \sqsubseteq Y$.

4. The 3 last properties are among the ones which make an extensional model of F complete for $F\eta$!!

5. They are **not far from webbed models of untyped λ-calculus,** so one could try to import some of the (few!) technologies discovered for them,

for example for comparing the theories of models.

for example for showing that the theory of the candidate model is included in $F\eta$.

12 Current and further work.

Continuous class :

- Is this class really able to represent many equational extensions of F ?
- or are most of its models complete either for F or $F\eta$?

Stable class.

Recall : the polymorphic maps are constrained, while they are not in the continuous case, in particular all the models are necessarily models of FC.

- still to be written down.
- Does it contain complete models for FC ? this would answer a problem of Longo related to the Longo-Milsted-Soloviev' s genericity theorem..
- richness of this class ?

Notions of models of F.

Try to express a still intuitive but more general notion of models.