Polymorphic Lambda Calculus and Subtyping

Adrian Fiech[1]
Computer Science Department
Memorial University of Newfoundland
St. John's, Canada A1B 3X5
fiech@cs.mun.ca

David A. Schmidt[2]
Dept. of Computing and Information Sciences
Kansas State University
Manhattan, Kansas 66506, USA
schmidt@cis.ksu.edu

Abstract

We present a denotational model for $F_{\sqsubseteq}$, the extension of second-order lambda calculus with subtyping defined in [Cardelli Wegner 1985]. Types are interpreted as arbitrary cpos and elements of types as natural transformations. We prove the soundness of our model with respect to the equational theory of $F_{\sqsubseteq}$ [Cardelli et al. 1991] and show coherence. Our model is of independent interest, because it integrates ad-hoc and parametric polymorphism in an elegant fashion, admits nontrivial records and record update operations, and formalizes an "order faithfulness" criterion for well-behaved multiple subtyping.

0 Introduction

A variety of denotational models for $F_{\sqsubseteq}$ (the second–order lambda calculus with subtyping and bounded quantification) [Cardelli Wegner 1985] have already been proposed [MacQueen et al. 1986, Bruce et al. 1990, Bruce Longo 1990, Amadio 1991, Cardone 1989, Abadi Plotkin 1990, Bruce Mitchell 1992, Breazu–Tannen et al. 1991], and this paper presents yet one more. Our reasons for doing so are threefold:

(i) We wish to formalize the intuition that both parametric and ad–hoc polymorphic functions are natural transformations and can live in the same framework. Real-life languages use ad-hoc polymorphism in crucial ways, but ad-hoc polymorphism is often prejudged as inelegant or undesirable. In his research on category-sorted algebra [Reynolds 1980], Reynolds points out that a well-behaved, ad-hoc polymorphic operation is a natural transformation in a specific sense, and one contribution of our work is to

[1]Partially supported by NSERC Grant OGP0170497

[2]Partially supported by NSF Grant CCR-9302962 and ONR Grant N00014–94–1–0866

show how simply well-behaved ad-hoc polymorphism can coexist with parameteric polymorphism.

(ii) We wish to allow models where the type expression $\forall\alpha.\alpha\to\alpha$ is interpreted as a set of mappings (natural transformations) that include more than merely the identity mapping (and its order–theoretic approximations). Our model is the first, to our knowledge, that has this trait, and therefore solves an outstanding problem with models for $F_{\sqsubseteq}$ noted by Bruce and Longo [Bruce Longo 1990] regarding record update operations. Briefly stated, an update operation on a record field, I, of type α, has the arity update_I : $\forall\tau\sqsubseteq\{I:\alpha\}.\tau\to\tau$. For earlier models of $F_{\sqsubseteq}$, the only function that update_I can denote is (an approximation of) the identity. But our model can and does admit non-identity, parametrically polymorphic denotations for update_I, including the desired updating function. Surprisingly, the modelling of records and updating is trivial once the model for $F_{\sqsubseteq}$ is formulated, so we proceed with a rigorous modelling of $F_{\sqsubseteq}$ and add records at the end of the paper.

(iii) We wish to develop the newly discovered notion of "functor order faithfulfulness" and study its role in the semantics of universal quantification. Order faithfulness is the formalization of well-behaved multiple subtyping. To explain, say that τ is a type that is an immediate subtype of the types $\tau_1, \tau_2, \ldots, \tau_n$, that is, $\tau = \sqcap\{\tau_1, \tau_2, \ldots, \tau_n\}$. For each τ_i, say that there is a coercion mapping, $coerce_i : \tau\to\tau_i$, which formalizes type-theoretic subsumption. The coercions are order faithful, if for all elements $a,b\in\tau$, for every τ_i, if $coerce_i(a)\sqsubseteq coerce_i(b)$, then $a\sqsubseteq b$ in τ as well. A functor that maps the ordering on type names to coercion mappings is order faithful if the coercions in its image are. Order faithfulness is crucial to constructing our model for $F_{\sqsubseteq}$ because it gives us guidance as to how to choose the proper coercions between types.

The end result is a novel model for $F_{\sqsubseteq}$ whose technical development is of interest in its own right.

The plan of the paper goes as follows: we begin with a survey of existing models for $F_{\sqsubseteq}$ and a short comparison of these models to our own. Next, we present the rudiments of category–sorted algebra, the foundation for our model, and we extend it for self–application. This yields a domain theory for second order lambda calculi, which we use to give a denotational semantics for $F_{\sqsubseteq}$. The semantics motivates a number of technical questions regarding our choices of categories and the functor order faithfulness property, which are answered. We finish with the extension of $F_{\sqsubseteq}$ to records.

Models for $F_{\sqsubseteq}$

We begin with a survey of previous models for polymorphic lambda calculus. To provide a proper comparison, for each model we describe:
(i) the types defined by the model;
(ii) the elements belonging to the types, most specifically, the elements of the type $\forall\alpha\sqsubseteq top.\alpha\to\alpha$;which we abbreviate below as $\forall\alpha.\alpha\to\alpha$.
(iii) the definition of subtype;
(iv) the semantics of terms, specifically, the denotation of the term $\lambda\alpha.\lambda x{:}\alpha.x$;
(v) coherence properties and additional interesting features of the model.

The ideal model

The ideal model of MacQueen, Plotkin, and Sethi [MacQueen et al.1986] provides denotations for types and terms of $F_{\sqsubseteq}$ but cannot handle equality laws. One begins by constructing the universal domain.

$$V = T + N + (V \to V) + (V \times V) + \ldots + \{wrong\}$$

where T is the cpo of truth values and N is the cpo of naturals.
(i) The types defined by the model are the ideals of V (down–closed sets closed under lubs of directed sets).
(ii) The elements of a type (ideal) are exactly the elements of the ideal. In particular, one interprets the type name $\forall\alpha.\sigma$ as $[\forall\alpha.\sigma]\rho = \bigcap_{i\in K}[\sigma]([\alpha\mapsto i]\rho)$, where K denotes the set of ideals that do not contain the element, wrong. Hence $[\forall\alpha.\alpha\to\alpha]\rho$ denotes the ideal of functions that map inputs from an ideal, i, to outputs in i; for the definition of V given above, this consists of approximations of the identity function on V.
(iii) The subtype relation is interpreted as subset, and coercion between types is embedding.
(iv) A term is interpreted in its "raw" (untyped) form as an element of V. Thus, the polymorphic identity term is interpreted as $[\![\lambda x.x]\!]\rho$, which is the identity function on V.
(v) The model possesses the following soundness property: $\Gamma\vdash e{:}\sigma \Rightarrow$ if $\rho\models\Gamma$ then $[\![e]\!]\rho\in[\sigma]\rho$. In addition, the model handles recursive types: for contractive functionals, the Banach fixed point theorem guarantees existence of a unique ideal defined by the functional. Unfortunatelly the ideal model is not a model for the second–order lambda calculus as the weak extensionality property fails [Bruce, et al. 1990].

PER models over ω

Here, one begins with the category, PER, of partial equivalence relations ("pers") over

the natural numbers, ω [Bruce, Longo 1990]. (Alternatively, one can work within the category of modest sets, of which PER is a subcategory.) Function application is defined by Goedel numbering: $m \bullet n = \phi_m(n)$.

(i) The types are exactly the pers on ω, and

(ii) the elements of a type (per), R, are those equivalence classes, $[n]_R$, such that n R n. In particular, we have that $[\sigma_1 \to \sigma_2]\rho = \{(m,n) \mid \forall p,q : p\ [\sigma_1]\rho\ q \text{ implies } m \bullet p\ [\sigma_2]\rho\ n \bullet q\}$ and also $[\forall \alpha \sqsubseteq \tau.\sigma]\rho = \Pi_{R \subseteq [\tau]\rho}[\sigma](\rho[\alpha \mapsto R])$, which in PER is isomorphic to $\bigcap_{R \subseteq [\tau]\rho}[\sigma](\rho[\alpha \mapsto R])$. This means that $[\forall \alpha.\alpha \to \alpha]\rho = \bigcap_{R \in PER} R \to R$. This type contains just one element: the equivalence class for the (Goedel numbers of) the identity function on ω.

(iii) The subtyping relation is the subset relation on objects of PER, thus, $\sigma_1 \sqsubseteq \sigma_2$ iff $[\sigma_1]\rho \subseteq [\sigma_2]\rho$, for all ρ. Note that, for types R_1 and R_2, $R_1 \subseteq R_2$ does not imply that the elements of type R_1 are a subset of the elements of the type R_2. Coercion of an element of type R_1 into an element of type R_2 is done with a function convert: $\forall \alpha.\forall \beta \sqsubseteq \alpha.\beta \to \alpha$. This means the usual subsumption rule,

$$\frac{\Sigma, \Gamma \vdash e : \sigma_1 \qquad \Sigma \vdash \sigma_1 \sqsubseteq \sigma_2}{\Sigma, \Gamma \vdash e : \sigma_2}$$

must be interpreted in terms of the convert operation.

(iv) A typed term, $\Sigma,\Gamma \vdash e:\sigma$, is given its semantics by induction on the derivation of its typing. In particular, $[\![\Sigma,\Gamma \vdash \lambda \alpha \sqsubseteq \sigma.e : \forall \alpha \sqsubseteq \sigma.\tau]\!]\rho = g$, where $g(R) = [\![\Sigma \cup \{\alpha \sqsubseteq \sigma\},\Gamma \vdash e:\tau]\!](\rho[\alpha \mapsto R])$, for $R \subseteq [\sigma]\rho$. As noted above, the interpretation of the polymorphic identity lambda abstraction must produce the identity map on PER.

(v) Because the semantics of a term is assigned within a second–order frame [Bruce et al. 1990], one obtains the desired coherence property: the denotation of a typed term is unaffected by the derivation of its typing. One does not obtain a "casting property", however: even when $\sigma_1 \sqsubseteq \sigma_2$ holds, $[\![\Sigma,\Gamma \vdash e:\sigma_1]\!]\rho \neq [\![\Sigma,\Gamma \vdash e:\sigma_2]\!]\rho$.

PER models over reflexive domains

The PER models developed independently by Amadio [Amadio 1991] and Cardone [Cardone 1989] replace ω by a "universal" domain, D, such as $D = N + B + (D \to D) + (D \times D)$.

(i) The types in the model are complete, uniform, partial equivalence relations, that is, pers that are directed complete, pointed (that is, $\perp$ R $\perp$), and uniform (that is, x R y implies $\pi_n(x)$ R $\pi_n(y)$, where π_n, $n \geq 0$, are the projection maps in the inverse limit construction of D).

(ii) Elements of a type are defined as in the PER–ω model, although the definition of D causes the type $[\forall\alpha\sqsubseteq ns.\alpha\to\alpha]\rho$ to contain (equivalence classes of) all order–theoretic approximations to the identity function on D.
(iii) Subtyping is defined as before.
(iv) A term is interpreted in its "raw" form: for typed term, $\Sigma,\Gamma\vdash e:\sigma$, we calculate $[\![erase\ e]\!]\rho$, where *erase* removes typing information from e, and we define $[\![\Sigma,\Gamma\vdash e:\sigma]\!]\rho = [\ [\![erase\ e]\!](erase\ \rho)\ ]_{[\sigma]\rho}$. In particular, $[\![\Sigma,\Gamma\vdash\lambda\alpha\sqsubseteq ns.\lambda x:\alpha.x : \forall\alpha\sqsubseteq ns.\alpha\to\alpha]\!]\rho = [id_D]_{[\forall\alpha\sqsubseteq ns.\alpha\to\alpha]\rho}$
(iv) The expected soundess property holds. A major achievement of this approach is the use of the complete uniform pers to model recursive types. The technique is a refinment of [Coppo 1985].

Abadi and Plotkin [Abadi, Plotkin 1990] extended the efforts by Amadio and Cardone. Their motivation was to obtain a complete partial ordering on the pers. This makes possible recursive types via inverse limit constructions on realizable functors and allows proofs by fixed–point induction. The authors work with pers that are complete, uniform, and additionally meet–closed and convex. (Meet closed: for sets X and Y indexed by I, if for all $i\in I$, $X_i\ R\ Y_i$, then $\sqcap X\ R\ \sqcap Y$. Convex: if $x\sqsubseteq y\sqsubseteq z$ and $x\ R\ z$, then $x\ R\ y$.) The universal domain, D, over which the pers are formed is required to be a Scott–domain. A consequence of these restrictions is that for $a,b\in D$, $a\sqsubseteq b$ implies $[a]_R\sqsubseteq_R[b]_R$ and that the elements of a type form a Scott–domain.

Bruce and Mitchell [Bruce Mitchell 1992] provide a general framework for the per–based models described above. Complete, uniform pers are called "nice". A collection, $\mathbb{R}$, of nice pers is acceptable if
(a) $\mathbb{R}$ contains the per $\{(\bot,\bot)\}$, and is closed under function space, records, and F–bounded quantification
(b) if $\{R^i \mid i\sqsubseteq\omega\}\subseteq\mathbb{R}$ and for all $j\geq i$, $R^j_{[i]} = R^i$, then there exists a unique $R'\in\mathbb{R}$ such that for all i, $R'_{[i]} = R^i$.
When working with acceptable pers, one can utilize a class of type functionals that is richer than that proposed by Abadi and Plotkin, allowing bounded and F–bounded polymorphism and giving models for F_ω with subtyping, bounded quantification, records and nontrivial recursive types. Examples of collections of acceptable pers are these constructed by Amadio, Cardone, and Abadi and Plotkin.

Explicit coercion models

A different approach to modeling $F_{\sqsubseteq}$ is taken by Breazu–Tannen, Coquand, Gunter and

Scedrov [Breazu–Tannen et al. 1991], where the "source language" of $F_{\sqsubseteq}$ plus records and recursive types is translated into a "target language" of explictly typed F_2 with recursive types, explicit coercion operations, and no subtyping relations. When one provides a model, such as a per model or a categorical model [Coquand et al. 1989], for the target language, one obtains a model via the translation for the source language. It is crucial that a model already exists for the target language.

(i) The types of the model are the types of the target language, and a type name, σ, in the source language is understood in terms of its translation, σ^*, a type in the target language. Examples are $(\sigma_1 \to \sigma_2)^* = \sigma_1^* \to \sigma_2^*$ and $(\forall \alpha \sqsubseteq \tau.\sigma)^* = \forall \alpha.(\alpha \to \tau^*) \to \sigma^*$. The latter states that bounded quantification is decoded into a function space that requires a coercion mapping as an argument to compute its result. In particular, $(\forall \alpha.\alpha \to \alpha)^*$ is $\forall \alpha.(\alpha \to ns^*) \to (\alpha^* \to \alpha^*)$.

(ii) The elements of a type in the target language are given, and for source language type, σ, its elements are those of σ^*.

(iii) Subtyping is modelled by assigning to each subtyping relation, $\sigma_1 \sqsubseteq \sigma_2$, in the source language, a coercion function, $f : \sigma_1^* \to \sigma_2^*$. We require that the coercion chosen for $\sigma \sqsubseteq \sigma$ is the identity on σ^*; because the relation, $\sqsubseteq$, is freely generated in the definition of $F_{\sqsubseteq}$, we obtain a coherence property for coercions, specifically: if $\sigma_1 \sqsubseteq \sigma_2 \sqsubseteq \sigma_3$, then $h : \sigma_1^* \to \sigma_3^* = (g: \sigma_2^* \to \sigma_3^*) \circ (f: \sigma_1^* \to \sigma_2^*)$.

(iv) Source language terms are interpreted in terms of their translations into the target language; the translation is defined by induction on the derivation of the typed source language term. The translation of the introduction rule for universal quantification

$$\frac{\Gamma, \alpha \sqsubseteq \tau \vdash e : \sigma}{\Gamma \vdash \lambda \alpha \sqsubseteq \tau.e \ : \ \forall \alpha \sqsubseteq \tau.\sigma}$$

is

$$\frac{\Gamma^*, \ \alpha : \mathrm{Type}, \ f : \alpha \to \tau^* \vdash e : \sigma^*}{\Gamma^* \ \vdash \lambda \alpha.\lambda f : \alpha \to \tau^*.e \ : \ \forall \alpha.(\alpha \to \tau^*) \to \sigma^*}$$

(v) The main result of this work is a coherence property: regardless of the way a well–typed source language term is derived, its semantics is unique. Because this approach "translates away" subtyping, and the target language model is given at the outset, the basic nature of bounded quantification is not directly addressed.

The model in this paper

The motivation for our model stems from category–sorted algebra [Reynolds 1980], where a programming language with subtyping is specified by a signature consisting of

a partial ordering, D, of sorts and a set, O, of operators. Each n–ary operator $f \in O$ is in fact a monotonic function $f: D^n \to D$. The intuition is that D specifies the subtyping of the programming language and each operator $f \in O$ is, to some degree, polymorphic on D. Typically, D possesses a topmost element, which stands for the "top", "nonsense" or "error" type. Following category–sorted algebra notation, we use "ns" to denote the topmost element. An interpretation, $\Im$, of D is a functor to a suitable category, e.g., $\Im: D \Rightarrow CPO_{\perp}$. Then, the operators are interpreted as natural transformations that respect $\Im$: for $f: D^n \to D$, f's interpretation must be a natural transformation $\eta_f: \Im^n(D^n) \dot{\to} \Im \circ f$. These ideas are developed formally in Section 2.

To model $F_{\sqsubseteq}$, we generalize category–sorted algebra to include operators that can self–apply.

(i) Types are elements from the reflexive domain $D \cong (B+(D \to D))^{ns}$, where B must be a finite Scott-domain. (The operators are taken from $f \in D \to D$.) Type names from $F_{\sqsubseteq}$ are mapped into types in D; the key mappings are:

$$[\sigma_1 \to \sigma_2]\rho = \lambda d \in D.\text{if } d \sqsubseteq [\sigma_1]\rho \text{ then } [\sigma_2]\rho \text{ else ns}$$

$$[\forall \alpha \sqsubseteq \tau.\sigma]\rho = \bigsqcap \{[\sigma](\rho[\alpha \mapsto d]) \mid d \sqsubseteq [\tau]\rho\}$$

The first clause exposes the usual subsumption principle for function type names, and the second emphasizes that a universally quantified type must behave consistently with respect to all of its instantiations. For example, $[\forall \alpha.\alpha \to \alpha]\rho = \bigsqcap \{[\alpha \to \alpha](\rho[\alpha \mapsto d]) \mid d \in D\} = \bigsqcap \{\lambda a \in D.\text{if } a \sqsubseteq d \text{ then a else ns} \mid d \in D\} = \lambda a \in D.a$

(ii) Next, the elements of a type are defined by the interpretation functor, $\Im: D \Rightarrow CPO_{\perp}$. The types in D are interpreted as cpos and the ordering on the types is interpreted as coercion maps. In particular, for $b \in B$, $\Im[b]$ is a base cpo (e.g, $\Im[Int] = Z$), and for $f: D \to D$, $\Im[f] = \Im \dot{\to} \Im \circ f$, that is, a cpo of natural transformations of "arity" f. The latter requires a proof that $\Im$ is well defined. (See page 12.) Thus, the elements of type $[\forall \alpha.\alpha \to \alpha]\rho$ are natural transformations of form $\Im \dot{\to} \Im$, which includes more than just approximations to the identity natural transformation.

(iii) Subtyping is defined by the ordering on D, and there is no requirement that for $d_1, d_2 \in D$, $d_1 \sqsubseteq d_2$ implies $\Im[d_1] \subseteq \Im[d_2]$. Since $\Im$ must be a functor, $\Im[d_1 \sqsubseteq d_2]$ defines a coercion morphism from $\Im[d_1]$ to $\Im[d_2]$ that all operations must respect. In this sense, coherence is "forced" into the semantics.

The new concept that is inherent to the model is that the set of coercion morphisms must be "order faithful": for every nonempty subset $S=\{d_1, d_2, \ldots, d_n\}$, $S \subseteq D$, and for the corresponding coercions, $f_i: \Im[\bigsqcap S] \to \Im[d_i]$, we must have the following: if, for every $a, b \in \Im[\bigsqcap S]$ $f_i(a) \sqsubseteq_{\Im[d_i]} f_i(b)$ holds for every $d_i \in S$, then $a \sqsubseteq_{\Im[\bigsqcap S]} b$ must hold also. That

is, if the images of a, $b \in \Im(|\ |S)$ are always ordered when they are coerced, then a and b are ordered in $\Im[\sqcap S]$ as well.
(iv) Terms in $F_{\sqsubseteq}$ are interpreted by the structure of their typing derivation. Most importantly, the natural transformation denoted by the term, $\Sigma,\Gamma \vdash \lambda\alpha\sqsubseteq\tau.M : \forall\alpha\sqsubseteq\tau.\sigma$, is defined uniquely as a consequence of the order faithfulness property mentioned above. In particular, $[\![\Sigma,\Gamma \vdash \lambda\alpha.\lambda x{:}\alpha.x : \forall\alpha.\alpha\to\alpha]\!]\rho = \eta$, where $\eta_d(v) = v$, for all $d \in D$.
(v) As noted earlier, the semantics is proved sound with respect to the interpretation of the type names, coherent, and satisfying the extensional equational theory of $F_{\sqsubseteq}$. The translation of type names into types looks superficially like that of the explicit coercion model, but a key difference is that the translation here inserts semantic content into type names rather than discarding it in favor of explicit coercions and a preassigned model.

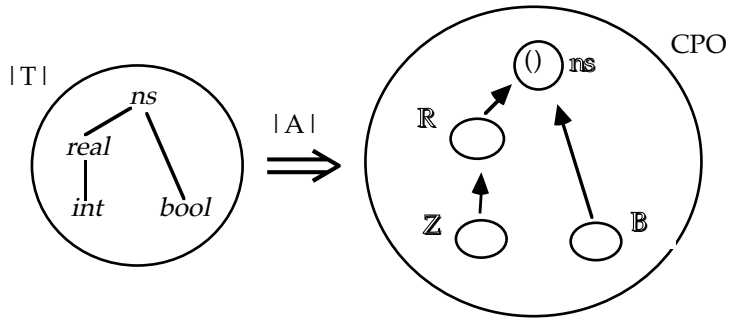
1 Category-Sorted Algebra

Category-sorted algebra was developed by Reynolds [Reynolds 1980] as a foundation for natural (in the category-theoretic sense) polymorphism in the presence of subtyping.

We motivate the topic with a quick example : for the data type names *int*, *real*, *bool*, and *ns* (which is the **nonsense** or **error** type), say that $int \sqsubseteq real$ and $t \sqsubseteq ns$, for all type names, *t*, and close the ordering under reflexivity and transitivity, giving a preordered set, |T|. Next, we desire a mixed mode addition operator, called *plus*. The interpretation of *plus* should respect the ordering in |T|, and we formalize this by giving *plus* a monotonic **typing function**, $T_{plus} : |T| \times |T| \to |T|$, which defines the type of answer *plus* produces from the types of its arguments. T_{plus} is defined :

$$T_{plus}(int,int) = int,$$
$$T_{plus}(int,real) = T_{plus}(real,int) = T_{plus}(real,real) = real,$$
$$T_{plus}(t_1,t_2) = ns, \text{ otherwise}$$

The typing function says that *plus* maps integer, integer pairs to integer answers; integer, real pairs to real answers; and so on.
Operation *plus* must behave as specified by T_{plus}, but it must also satisfy coherence with respect to $\sqsubseteq$; this is formalized by interpreting *plus* as a natural transformation. First, the functor, |A| : |T| $\Rightarrow$ CPO associates each type name to its meaning set, e.g. $|A|(int) = \mathbb{Z}$, $|A|(real) = \mathbb{R}$, etc, and the functor maps the subtyping ordering into a family of coercion maps, e.g., $|A|(int \sqsubseteq real)$ is a coercion map from $\mathbb{Z}$ to $\mathbb{R}$. We have:



Thus *plus* is interpreted as a natural transformation,

$$A^{plus}: |A| \times |A| \dashrightarrow |A| \circ T_{plus}.$$

Here is the expected definition of A^{plus} :

$$A^{plus}_{int,int} = \text{addition on } \mathbb{Z}$$
$$A^{plus}_{int,real} = A^{plus}_{real,int} = \text{mixed mode addition on } \mathbb{Z} \text{ and } \mathbb{R}$$
$$A^{plus}_{real,real} = \text{addition on } \mathbb{R}$$
$$\text{otherwise, } A^{plus}_{t_1,t_2} = f, \text{ where } f(v_1,v_2) = () \in \mathbb{ns}.$$

(For a natural transformation η from F to G, η_d denotes the morphism from F[d] to G[d])

Coercions can interact with the additions without affecting the final result, e.g., for $v_1, v_2 \in \mathbb{Z}$, $|A|(int \sqsubseteq real)(A^{plus}_{int,int}(v_1, v_2)) = A^{plus}_{real,real}(|A|(int \sqsubseteq real)v_1, |A|(int \sqsubseteq real)v_2)$. That is, two integers can be added and the result coerced to a real or two integers can be coerced to reals and then added — the result is the same.

Here are the fundamental definitions.

Let C = (Ob(C), Mor(C), id_C, $\circ_C$) be a category. Assume that C has finite products.

Definition 1.1

An **Ω-signature** is a pair, **(Ω, ar)**, where Ω is a set of operator names and $ar: \Omega \to \mathbb{N}$ is a function that gives the arity of the operators.

Definition 1.2

An **Ω-algebra** (based on C) is a pair, $T = \mathbf{(|T|, \{ T_\omega \mid \omega \in \Omega \})}$, where $|T| \in Ob(C)$ is the carrier and, for each $\omega \in \Omega$, $T_\omega : |T|^{ar(\omega)} \to |T| \in Mor(C)$ is an ar(ω)–ary operation on |T|.

Definition 1.3

An **Ω-T-signature** is a triple, **(Ω, ar, T)**, where Ω and ar are as before, and T is an Ω-algebra based on PreO, the category of preordered sets.

An Ω-T-signature specifies a set of polymorphic operators and their typing functions, e.g., *plus* ∈ Ω, ar(*plus*) = 2, |T| is the poset of types seen in the previous example, and T_{plus} is the typing function for *plus*.

Definition 1.4

An **Ω-T-category-sorted algebra** (based on C) is a pair,

$$A = (|A|, \{ A^{\omega} \mid \omega \in \Omega \}),$$

where $|A| : |T| \Rightarrow C$, the carrier of A, is a functor, and for each $\omega \in \Omega$, operation $A^{\omega} : |A|^{ar(\omega)} \dashrightarrow |A| \circ T_{\omega}$ is a natural transformation.

The associated machinery – homomorphisms, freely generated algebras, and so on – is presented in [Reynolds 1980].

2 Self-Application

One obvious generalization of the constructions in the previous section is replacing PreO, the category of preordered sets, in Definition 1.3 by $CPO_{\perp}$, the category of complete partially ordered sets with ⊥. Another is allowing |T| to be defined with domain constructions such as ×, + and → . For example, if B={*int*,*real*,…} is a set of base type names, then define $|T| \cong (B + (B \to B))^{ns}$. (Here $(D)^{ns}$ denotes the cpo D with an additional top element ns). $B \to B$ is the collection of typing functions of polymorphic operations on B, and for $f : B \to B$, $|A|(f) : F_B \dashrightarrow F_B \circ f$ is the collection of natural transformations, ordered pointwise, that have f as their typing function. ($F_B : B \Rightarrow CPO_{\perp}$ is the interpretation functor for base types.)

The next generalization is to incorporate the entire hierarchy of typing functions for polymorphic operations into |T|. We define:

$$|T| \cong (B + (|T| \to |T|))^{ns}$$

We will use elements of |T| as the "types" of phrases in $F_{\sqsubseteq}$

|T|'s definition can be solved by the usual techniques, e.g., [Smyth and Plotkin 1982], so the interesting part is the definition of the interpretation functor $|A| : |T| \Rightarrow CPO_{\perp}$. For clarity, these particular instances of |T| and |A| will be denoted as $D_{\mathfrak{S}}$ and $\mathfrak{S}$, respectively.

$\Im$ must be a solution to this recursive definition.

$\Im[\bot] = \{\bot\}$
$\Im[ns] = \{()\}$
$\Im[b \in B] = F_B(b)$, where $F_B : B \Rightarrow CPO_\bot$ interprets base types
$\Im[f \in |T| \to |T|] = \Im \dashrightarrow \Im \circ f$
$\Im[\bot \sqsubseteq d] = \lambda x \in \Im[\bot].\bot_{\Im[d]}$
$\Im[d \sqsubseteq ns] = \lambda x \in \Im[d].()$
$\Im[b \sqsubseteq b'] = F_B[b \sqsubseteq b']$
$\Im[f \sqsubseteq f'] = \lambda \eta \in \Im \dashrightarrow \Im \circ f.(\lambda d \in |T| . \Im[f(d) \sqsubseteq f'(d)] \circ \eta_d)$
(η_d denotes the morphism from $\Im[d]$ to $\Im \circ f[d]$)

Elements in the set $\Im \dashrightarrow \Im \circ f$ (we use η, κ, γ for them) will be used as the denotations of those lambda abstractions with type f in $F_\sqsubseteq$.

Since $\Im$ is defined recursively we are not guaranteed that such a functor exists. Fortunately, we can locate a solution in the category $Func_{CPO_\bot}$.

Definition 2.1

The category $\mathbf{Func_{CPO_\bot}}$ has as its objects pairs ($D_F \in Ob(CPO_\bot)$, $F : D_F \Rightarrow CPO_\bot$), such that for functor F, all $F[d \sqsubseteq d']$ are strict functions and $F[\bot] \cong \{\bot\}$. The morphisms between two objects (D_F,F) and (D_G,G) are pairs ($f : D_F \to D_G, \eta : F \dashrightarrow G \circ f$), where f is strict. Composition on morphisms ($f : D_F \to D_G$, $\eta : F \dashrightarrow G \circ f$) and ($g : D_G \to D_H$, $\gamma : G \dashrightarrow H \circ g$) is defined as ($g \circ f$, $\lambda d \in D_F . \gamma_{f(d)} \circ \eta_d$).

Lemma 2.2

Let $F, G : D_F \to CPO_\bot$ be functors. The set $\mathbf{F \dashrightarrow G}$ of natural transformations between F and G, equipped with the canonical order $\eta \sqsubseteq \gamma$ iff $\forall d \in D_F : \eta_d \sqsubseteq \gamma_d$, is a cpo with $(\hat{\bigsqcup}\{\eta^n\})_d = \hat{\bigsqcup}\{\eta^n{}_d\}$. ($\eta_d$ resp. γ_d are morphism from F[d] to G[d])

❑

The category $Func_{CPO_\bot}$ is closed under the common domain–theoretical operations, which are defined in the following.

Definition 2.3

Let $F, G \in Func_{CPO_\bot}$.

a) The **nonsense functor** $ns : Func_{CPO_\bot} \to Func_{CPO_\bot}$ is defined as
(use F^{ns} for ns(F)) :
$F^{ns} : (D_F)^{ns} \to CPO_\bot$

$F^{ns}[ns] = \{()\}$
$F^{ns}[d \in D_F] = F[d]$
$F^{ns}[d \sqsubseteq ns] = \lambda x \in F[d].()$
$F^{ns}[d \sqsubseteq d'] = F[d \sqsubseteq d']$
$ns(f : D_F \to D_G, \eta : F \dashrightarrow G \circ f)$
$= (\lambda x \in (D_F)^{ns}.\text{if } x=ns \text{ then } ns \text{ else } f(x),\ \ \lambda d \in (D_F)^{ns}.\text{if } d=ns \text{ then } \lambda x \in \{()\}.() \text{ else } \eta_d)$

b) The **product functor** $\times : Func_{CPO_\perp} \times Func_{CPO_\perp} \to Func_{CPO_\perp}$ is defined as :
$F \times G : D_F \times D_G \to CPO_\perp$
$F \times G[(d_F, d_G)] = F[d_F] \times G[d_G]$
$F \times G[(d_F \sqsubseteq d_F', d_G \sqsubseteq d_G')] = (F[d_F \sqsubseteq d_F'],\ \ G[d_G \sqsubseteq d_G'])$
$\times((f,\eta),(g,\gamma)) = (f \times g,\ \eta \times \gamma)$, where $\eta \times \gamma := \lambda(d_F, d_G) \in D_F \times D_G.\eta_{d_F} \times \gamma_{d_G}$

c) The **sum functor** $+ : Func_{CPO_\perp} \times Func_{CPO_\perp} \to Func_{CPO_\perp}$ is defined as :
$F+G : D_F + D_G \to CPO_\perp$
$F+G[\perp] = \{\perp\}$
$F+G[(1,d_F)] = F[d_F]$
$F+G[(2,d_G)] = G[d_G]$
$F+G[\perp \sqsubseteq (1,d_F)] = \lambda x \in \{\perp\}.\ \perp_{F[d_F]}, \qquad F+G[\perp \sqsubseteq (2,d_G)] = \lambda x \in \{\perp\}.\ \perp_{G[d_G]}$
$F+G[(1,d_F) \sqsubseteq (1,d_F')] = F[d_F \sqsubseteq d_F'], \qquad F+G[(2,d_G) \sqsubseteq (2,d_G')] = G[d_G \sqsubseteq d_G']$
$+((f,\eta),(g,\gamma)) = (f+g,\ \eta+\gamma)$,
where $\eta+\gamma := \lambda x \in D_F + D_G.\text{cases } x \text{ of } \perp \to \lambda x \in \{\perp\}.\perp \mid (1,d_F) \to \eta_{d_F} \mid (2,d_G) \to \gamma_{d_G}$.

d) The **exponentiation functor** $\to : Func_{CPO_\perp}{}^{op} \times Func_{CPO_\perp} \to Func_{CPO_\perp}$ is defined:
$F \to G : [D_F \to D_G] \to CPO_\perp$
$F \to G[f] = F \dashrightarrow G \circ f$
$F \to G[f \sqsubseteq f'] = \lambda \eta \in F \dashrightarrow G \circ f.(\lambda d \in D_F.G[f(d) \sqsubseteq f'(d)] \circ \eta_d)$
$\to((f,\eta),(g,\gamma)) \ = \ (\lambda h \in [D_F \to D_G].g \circ h \circ f,\ \lambda h \in [D_F \to D_G].\lambda \phi \in F \dashrightarrow G \circ h.\gamma \circ \phi \circ \eta)$

We use these functors to define the functor $\mathcal{K} : Func_{CPO_\perp} \Rightarrow Func_{CPO_\perp}$ with $\mathcal{K}[F] = (F_B + (F \to F))^{ns}$. Here $F_B : B \Rightarrow CPO_\perp$ is the functor that assigns the appropriate cpos and coercion morphisms to the types in B. Now we officialy define our desired functor, $\mathfrak{I}$.

Definition 2.4

Let B be a cpo of primitive types and let $F_B : B \Rightarrow CPO_\perp$ be a functor in $Func_{CPO_\perp}$. Furthermore define the functor $\mathcal{K} : Func_{CPO_\perp} \Rightarrow Func_{CPO_\perp}$ as $\mathcal{K}[F] = (F_B + (F \to F))^{ns}$. The **functor** $\mathfrak{I} : D_{\mathfrak{I}} \Rightarrow CPO_\perp$, with $D_{\mathfrak{I}} \cong (B + (D_{\mathfrak{I}} \to D_{\mathfrak{I}}))^{ns}$, is the initial fixed point of $\mathcal{K}^E$ constructed in the category $Func_{CPO_\perp}{}^E$ (where the morphisms are embedding–projection pairs; see

[Smyth and Plotkin 1982]).

For this definition to be correct we must be able to construct the desired fixed point of $\mathcal{K}$. We must show that $\mathrm{Func}_{\mathrm{CPO}_\perp}$ is an O–category [Smyth and Plotkin 1982] that has limits of ω^{op}–chains. Then $\mathrm{Func}_{\mathrm{CPO}_\perp}{}^{E}$ is an ω–category. It is also necessary to prove that the functors $+, \rightarrow$ and ns on $\mathrm{Func}_{\mathrm{CPO}_\perp}$ are locally continuous.

Proposition 2.5 [Schmidt 1990]

The category $\mathrm{Func}_{\mathrm{CPO}_\perp}$ is an O–category, where the order on $\hom(F : D_F \Rightarrow \mathrm{CPO}_\perp, G : D_G \Rightarrow \mathrm{CPO}_\perp)$ is defined as follows :

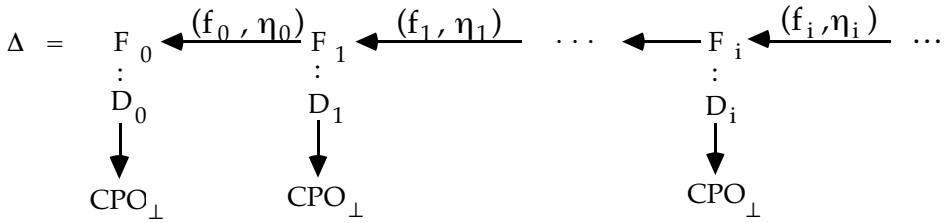
$$(f,\eta) \sqsubseteq (f',\eta') \Leftrightarrow f \sqsubseteq f' \text{ and for all } d \in D_F : G[f(d) \sqsubseteq f'(d)] \circ \eta_d \sqsubseteq \eta'_d$$

The terminal object is the functor $\perp : \{\perp\} \Rightarrow \mathrm{CPO}_\perp$, every hom–set has a least element, and composition of morphisms is left–strict. Also, the category $\mathrm{Func}_{\mathrm{CPO}_\perp}$ has all limits of ω^{op}–chains.

Proof

The proof is straightforward, so we show here only that all ω^{op}–chains have limits.

Let Δ be an ω^{op}–chain in $\mathrm{Func}_{\mathrm{CPO}_\perp}$:



The limiting functor $F_\omega : D_\omega \Rightarrow \mathrm{CPO}_\perp$ is constructed as follows. The domain D_ω is the limit of the D_i's: $D_\omega = \{(d_0,d_1,\ldots,d_n,\ldots) \mid d_n = f_n(d_{n+1})\}$. $F_\omega[(d_0,d_1,\ldots,d_n,\ldots)] = \{(a_0,a_1,\ldots,a_n,\ldots) \in \Pi_{n\geq 0}(F[d_n]) \mid a_n = (\eta_n)_{d_{n+1}}(a_{n+1})\}$. $F_\omega[(d_0,\ldots,d_n,\ldots) \sqsubseteq (d_0',\ldots,d_n',\ldots)] = \Pi_{n\geq 0}(F_n[d_n \sqsubseteq d_n'])$. It is clear that F_ω is a functor. The limiting morphisms $(h_n,\kappa_n) : F_\omega \rightarrow F_n$ are defined as : $h_n((d_0,\ldots,d_n,\ldots)) = d_n$ and $(\kappa_n)_{(d_0,\ldots,d_n,\ldots)}((a_0,\ldots,a_n,\ldots)) = a_n$. F_ω together with morphisms (h_n,κ_n) is a natural source. Let now $(G : D_G \Rightarrow \mathrm{CPO}_\perp, (g_n,\gamma_n) : G \rightarrow F_n)$ be another natural source. The unique mediating morphism $(r,\rho) : G \rightarrow F_\omega$ is given by : $r(x) = \Pi_{n\geq 0}(g_n(x))$ and $\rho_x(a) = \Pi_{n\geq 0}((\gamma_n)_x(a))$.

❑

Proposition 2.6 [Schmidt 1990]

The functors from Definition 2.3 are locally continuous.

❑

So, we can work with the category $Func_{CPO_\perp}^{E}$ and with the functor $\mathcal{K}^E : Func_{CPO_\perp}^{E} \Rightarrow Func_{CPO_\perp}^{E}$. The initial fixed point of $\mathcal{K}^E$ can be now constructed.

Now that these technical hurdles are surmounted, we can define operators that self–apply. For an operator, ω, the typing function, T_ω, is an element of $|T|$, and the operation, A^ω, is an element of the cpo, $\Im[T_\omega] = \Im \dashrightarrow \Im \circ T_\omega$. Hence, $A^\omega_{T_\omega} : \Im[T_\omega] \rightarrow \Im[T_\omega(T_\omega)]$ is a function that can take A^ω as its argument.

3 A model for second–order lambda calculus with inheritance

In this section we develop a model for an extension of system F, called $F_{\sqsubseteq}$, as described in [Cardelli et al 1991]. Additionally we include type and term constants. The reader interested in properties of $F_{\sqsubseteq}$ should consult the above paper. There it is shown that although the equational theory of $F_{\sqsubseteq}$ is not conservative over F (because of rule Eq appl2), the typing theory of $F_{\sqsubseteq}$ is conservative over F and F_1 typing (in the later case "modulo an equality"). The reader can also find examples how a range of common constructs can be encoded in $F_{\sqsubseteq}$.

When introducing the model we hold the technical aspects to a minimum and delay the in–depth treatment to the next section.

The syntax of our calculus follows.

Definition 3.1

The **type names** of the calculus $F_{\sqsubseteq}$ are defined by the following abstract grammar

$$\mathbb{T} = \mathbb{V} \mid \mathbb{C} \mid \mathbb{T} \rightarrow \mathbb{T} \mid \forall \mathbb{V} \sqsubseteq \mathbb{T}.\mathbb{T} \mid$$

where $\mathbb{V}$ resp. $\mathbb{C}$ are countable sets of type variables resp. type constants with $ns \in \mathbb{C}$.
We will use α,β for elements of $\mathbb{V}$, t,s for elements of $\mathbb{C}$ and σ,τ,ξ,ζ for elements of $\mathbb{T}$.
Polymorphic types of the form $\forall\alpha.\sigma$ are represented by bounded polymorphic types $\forall\alpha \sqsubseteq ns.\sigma$.

Definition 3.2

The set of untyped **$F_{\sqsubseteq}$–terms** Λ is defined by the following abstract syntax

$$\Lambda = V \mid C \mid \Lambda\Lambda \mid \lambda V:\mathbb{T}.\Lambda \mid \lambda\mathbb{V} \sqsubseteq \mathbb{T}.\Lambda$$

where V resp. C are countable sets of term variables resp. term constants.
We will use x,y for elements of V, c for elements of C and M,N and Q for elements of Λ.

Definition 3.3

Let $\Sigma^{\mathbf{C}}$ be a partial ordering on the type constants. Furtheremore let Σ be a set of subtyping assumptions of the form $\alpha \sqsubseteq \xi$. A subtyping rule $\sigma \sqsubseteq \tau$ is derivable from Σ, notation $\Sigma \vdash \sigma \sqsubseteq \tau$, if $\Sigma \vdash \sigma \sqsubseteq \tau$ can be produced by the following rules :

(Sub ns) $\quad \Sigma \vdash \sigma \sqsubseteq ns$

(Sub const) $\quad \Sigma \vdash t_1 \sqsubseteq t_2 \qquad$ for $t_1 \sqsubseteq t_2 \in \Sigma^{\mathbf{C}}$

(Sub refl) $\quad \Sigma,\ \alpha \sqsubseteq \xi \vdash \alpha \sqsubseteq \xi$

(Sub trans) $\quad \dfrac{\Sigma \vdash \sigma \sqsubseteq \tau \quad \Sigma \vdash \tau \sqsubseteq \xi}{\Sigma \vdash \tau \sqsubseteq \xi}$

(Sub $\rightarrow$) $\quad \dfrac{\Sigma \vdash \sigma_1 \sqsubseteq \sigma_2 \quad \Sigma \vdash \tau_1 \sqsubseteq \tau_2}{\Sigma \vdash \sigma_2 \rightarrow \tau_1 \sqsubseteq \sigma_1 \rightarrow \tau_2}$

(Sub $\forall$) $\quad \dfrac{\Sigma \vdash \sigma_1 \sqsubseteq \sigma_2 \quad \Sigma, \alpha \sqsubseteq \sigma_1 \vdash \tau_1 \sqsubseteq \tau_2}{\Sigma \vdash (\forall \alpha \sqsubseteq \sigma_2 . \tau_1) \sqsubseteq (\forall \alpha \sqsubseteq \sigma_1 . \tau_2)} \quad \alpha \notin FV(\Sigma)$

A statement in $F_{\sqsubseteq}$ is of the form $M : \sigma$ with $M \in \Lambda$ and $\sigma \in \mathbb{T}$. A set of statements $\Gamma = \{x_1 : \sigma_1, \ldots, x_m : \sigma_m\}$ with distinct $x_i \in V$ is called a basis.

Definition 3.4

A statement $M : \sigma$ is derivable in the calculus from a basis Γ and subtyping assumptions Σ, notation $\Sigma, \Gamma \vdash M : \sigma$, if $\Sigma, \Gamma \vdash M : \sigma$ can be produced by the following assignments rules

(const) $\quad \Sigma, \Gamma \vdash c^{\tau} : \tau \quad$ for all $c^{\tau} \in C$

(var) $\quad \Sigma, \Gamma \vdash x : \sigma \quad$ if $x : \sigma \in \Gamma$

($\rightarrow$E) $\quad \dfrac{\Sigma, \Gamma \vdash M : \sigma \rightarrow \tau \quad \Sigma, \Gamma \vdash N : \sigma}{\Sigma, \Gamma \vdash MN : \tau}$

($\rightarrow$I) $\quad \dfrac{\Sigma, \Gamma, x : \sigma \vdash M : \tau}{\Sigma, \Gamma \vdash \lambda x : \sigma . M : \sigma \rightarrow \tau}$

($\forall$E) $\quad \dfrac{\Sigma, \Gamma \vdash M : \forall \alpha \sqsubseteq \tau . \sigma \quad \Sigma \vdash \xi \sqsubseteq \tau}{\Sigma, \Gamma \vdash M\xi : \sigma[\alpha := \xi]}$

(∀I) $$\frac{\Sigma,\alpha\sqsubseteq\tau,\Gamma\vdash M:\sigma}{\Sigma,\Gamma\vdash\lambda\alpha\sqsubseteq\tau.M:\forall\alpha\sqsubseteq\tau.\sigma}\qquad \alpha\notin(FV(\Gamma)\cup FV(\Sigma))$$

(Inh) $$\frac{\Sigma,\Gamma\vdash M:\sigma\qquad\Sigma\vdash\sigma\sqsubseteq\tau}{\Sigma,\Gamma\vdash M:\tau}$$

$\sigma[\alpha:=\xi]$ is the result of substituting ξ for the free occurrences of α in σ.

Definition 3.5

An equation in $F_{\sqsubseteq}$ is of the form $\Sigma,\Gamma\vdash M\leftrightarrow N:\sigma$, where $\Sigma,\Gamma\vdash M:\sigma$ and $\Sigma,\Gamma\vdash N:\sigma$. The equational theory of $F_{\sqsubseteq}$ is defined by the following set of inference rules :

(Eq symm) $$\frac{\Sigma,\Gamma\vdash M\leftrightarrow N:\sigma}{\Sigma,\Gamma\vdash N\leftrightarrow M:\sigma}$$

(Eq trans) $$\frac{\Sigma,\Gamma\vdash M\leftrightarrow N:\sigma\qquad\Sigma,\Gamma\vdash N\leftrightarrow Q:\sigma}{\Sigma,\Gamma\vdash M\leftrightarrow Q:\sigma}$$

(Eq x) $$\frac{\Sigma,\Gamma\vdash x:\sigma}{\Sigma,\Gamma\vdash x\leftrightarrow x:\sigma}$$

(Eq collapse) $$\frac{\Sigma,\Gamma\vdash M:ns\qquad\Sigma,\Gamma\vdash N:ns}{\Sigma,\Gamma\vdash M\leftrightarrow N:ns}$$

(Eq fun) $$\frac{\Sigma,\Gamma,x:\sigma\vdash M\leftrightarrow N:\tau}{\Sigma,\Gamma\vdash\lambda x:\sigma.M\leftrightarrow\lambda x:\sigma.N:\sigma\to\tau}$$

(Eq appl) $$\frac{\Sigma,\Gamma\vdash M_1\leftrightarrow M_2:\sigma\to\tau\qquad\Sigma,\Gamma\vdash N_1\leftrightarrow N_2:\sigma}{\Sigma,\Gamma\vdash M_1N_1\leftrightarrow M_2N_2:\tau}$$

(Eq fun2) $$\frac{\Sigma,\alpha\sqsubseteq\sigma,\Gamma\vdash M\leftrightarrow N:\tau}{\Sigma,\Gamma\vdash\lambda\alpha\sqsubseteq\sigma.M\leftrightarrow\lambda\alpha\sqsubseteq\sigma.N:\forall\alpha\sqsubseteq\sigma.\tau}$$

(Eq appl2) $$\frac{\Sigma,\Gamma\vdash M\leftrightarrow N:\forall\alpha\sqsubseteq\sigma.\tau\quad\Sigma\vdash\xi_1\sqsubseteq\sigma\quad\Sigma\vdash\xi_2\sqsubseteq\sigma\quad\Sigma\vdash\tau[\alpha:=\xi_1]\sqsubseteq\zeta\quad\Sigma\vdash\tau[\alpha:=\xi_2]\sqsubseteq\zeta}{\Sigma,\Gamma\vdash M\xi_1\leftrightarrow N\xi_2:\zeta}$$

(Eq eta) $$\frac{\Sigma,\Gamma\vdash M\leftrightarrow N:\sigma\to\tau}{\Sigma,\Gamma\vdash\lambda y:\sigma.(My)\leftrightarrow N:\sigma\to\tau}\qquad y\notin dom(\Gamma)$$

(Eq eta2) $$\frac{\Sigma,\Gamma\vdash M\leftrightarrow N:\forall\alpha\sqsubseteq\sigma.\tau}{\Sigma,\Gamma\vdash\lambda\beta\sqsubseteq\sigma.(M\beta)\leftrightarrow N:\forall\alpha\sqsubseteq\sigma.\tau}\qquad \beta\notin dom(\Sigma)\cup dom(\Gamma)$$

(Eq beta) $$\frac{\Sigma,\Gamma,x:\sigma \vdash M \leftrightarrow N:\tau \quad \Sigma,\Gamma \vdash P \leftrightarrow Q:\sigma}{\Sigma,\Gamma \vdash (\lambda x:\sigma.M)P \leftrightarrow N[x:=Q]:\tau}$$

(Eq beta2) $$\frac{\Sigma,\alpha \sqsubseteq \sigma,\Gamma \vdash M \leftrightarrow N:\tau \quad \Sigma \vdash \xi \sqsubseteq \sigma}{\Sigma,\Gamma \vdash (\lambda \alpha \sqsubseteq \sigma.M)\xi \leftrightarrow N[\alpha:=\xi]:\tau[\alpha:=\xi]}$$

Intuitively the equation $\Sigma,\Gamma \vdash M \leftrightarrow N:\sigma$ means that the terms M and N describe the same element of the type corresponding to σ.

The category $Func_{CPO_\perp}$ is the essential part of our model. In this category we construct the functor $\Im$, which is a solution to the functional defined in the previous section. The functor $\Im: D_\Im \Rightarrow CPO_\perp$ maps elements in $D_\Im$ to their corresponding cpos, where $D_\Im \cong (B+(D_\Im \to D_\Im))^{ns}$ and B is the domain of base types. For any derivable statement $\Sigma,\Gamma \vdash M:\sigma$ we calculate its semantics in two major steps. First, we find the type, $f_\sigma \in D_\Im$, associated with the type name $\sigma \in \mathbb{T}$. Next, we calculate the element in the cpo $\Im[f_\sigma]$ that contains the meaning of M. We do this with the semantics functions in Definition 3.13. To calculate f_σ from σ, we use a function, [_], which assigns to each type name, σ, a type, $[\sigma]\rho$, in $D_\Im$.

Definition 3.6

Let $\mathbb{T}$ be the type name system and let $\mathbb{V}$ be a countable set of type variables. A **type environment** is a function $\rho: \mathbb{V} \to D_\Im$. The **empty** type environment is the function $\lambda v \in \mathbb{V}.\perp$. $\mathbf{Env_{type}}$ is the set of all type environments.

Definition 3.7

The function $[_]: \mathbb{T} \times Env_{type} \to D_\Im$ is given by :

a) $[t \in \mathbb{C}]\rho = t_B$, where $t_B \in B$.
b) $[\alpha \in \mathbb{V}]\rho = \rho(\alpha)$
c) $[\sigma \to \tau]\rho = \lambda d \in D_\Im$. if $d \sqsubseteq [\sigma]\rho$ then $[\tau]\rho$ else *ns*.
d) $[\forall \alpha \sqsubseteq \tau.\sigma]\rho = \bigsqcap_{d \in D_\Im, d \sqsubseteq [\tau]\rho} \{[\sigma]([\alpha \mapsto d]\rho)\}$

Notice that $[\sigma[\alpha:=\tau]]\rho = [\sigma]([\alpha \mapsto [\tau]\rho]\rho)$. Also, in case (c) the function $[\sigma \to \tau]\rho$ is continuous and therefore is indeed in $[D_\Im \to D_\Im]$. For case (d) to be correct, infima of nonempty sets must exist in $D_\Im$. (We will require that $D_\Im$ is bounded complete.)
Cases (a) and (b) are easy to understand : for each type constant we have a corresponding primitive type in the set B, and to evaluate a type variable we look it up in the environment.

In case (c) it is not enough to require that the resulting type, a function, maps $[\sigma]\rho$ to $[\tau]\rho$ and everything else to *ns*. First, this function would not be continuous and thus not in $D_{\Im}$, and second, if a lambda term accepts an argument of type $[\sigma]\rho$ and maps it to some element of type $[\tau]\rho$, it also should accept all arguments of type d such that $d \sqsubseteq [\sigma]\rho$ and map them to answers in $[\tau]\rho$. For example, the type $[real \to bool]\rho$ is:

real	→	*bool*
int	→	*bool*
$\perp$	→	*bool*
else	→	*ns*.

Case (c) also justifies calling the elements in $\mathbb{T}$ *type names* instead of *types*. If for some reason we decide that *int* is not a subtype of *real*, then the type of real → bool (which is a type name) changes to a function which maps *int* to *ns*. So, the type name *real* → *bool* can have different types, depending on the ordering on the set B.

Case (d) needs a little more explanation. In Definition 3.13, we will see that statement $\Sigma,\Gamma \vdash M: \forall \alpha \sqsubseteq \tau.\sigma_1 \to \sigma_2$ denotes a polymorphic function, that is, a natural transformation $\eta \in \Im[\forall \alpha \sqsubseteq \tau.\sigma_1 \to \sigma_2]\rho$. We define the natural transformations $\eta^t \in \Im[\sigma_1 \to \sigma_2]([\alpha \mapsto t]\rho)$ as the denotation of the statement $\Sigma,\Gamma \vdash M: \sigma_1 \to \sigma_2$ evaluated in the type environment $[\alpha \mapsto t]\rho$, for $t \sqsubseteq [\tau]\rho$. The family η^t, $t \sqsubseteq [\tau]\rho$, must behave consistently, that is, if η^t is applied to some argument x in a given context we should get a result consistent with that obtained by applying η to x in the same context : $\ldots\eta(x)\ldots \cong \ldots\eta^t(x)\ldots$. Therefore for each type $d \in D_{\Im}$, whenever $d' = ([\sigma_1 \to \sigma_2]([\alpha \mapsto t]\rho))(d)$, η must map arguments from $\Im[d]$ to elements in a subtype of $\Im[d']$. Otherwise, η would produce a result that cannot be dealt with. (We must be able to transform results produced by η into results produced by η^t.) So the type of the type name $\forall \alpha \sqsubseteq \tau.\sigma_1 \to \sigma_2$ must be less than all the types $([\sigma_1 \to \sigma_2]([\alpha \mapsto t]\rho))$, for all $t \sqsubseteq [\tau]\rho$.

The environment $\rho \in Env_{type}$ satisfies Σ, written $\rho \models \Sigma$, iff for all $\alpha \sqsubseteq \xi \in \Sigma$, $\rho(\alpha) \sqsubseteq [\xi]\rho$.

Lemma 3.8

Let $\Sigma \vdash \sigma \sqsubseteq \tau$. For all $\rho \models \Sigma$, $[\sigma]\rho \sqsubseteq [\tau]\rho$.

❑

The term environment is defined similarly to the type environment.

Definition 3.9

Let V be the set of term variables. A **term environment** is a function $\rho: V \to \bigcup_{d \in D_{\Im}} \Im[d]$. The **empty** term environment is a function $\lambda v \in V.\perp_{\Im[\perp]}$. $\mathbf{Env_{term}}$ is the set of all term

environments.

Definition 3.10

The set of **$F_{\sqsubseteq}$–environments**, **$Env_{F_{\sqsubseteq}}$**, is defined as $Env_{F_{\sqsubseteq}} = Env_{term} \times Env_{type}$.

The **empty** $F_{\sqsubseteq}$–environment, $\rho_{\emptyset}$, is the product of the empty term and type environments.

To simplify notation we will apply $\rho \in Env_{F_{\sqsubseteq}}$ to both type variables and term variables, instead of using the product $(\rho_{term}, \rho_{type})$.
When we evaluate a statement $\Sigma, \Gamma \vdash M : \sigma$, the $F_{\sqsubseteq}$–environment ρ must satisfy Γ and Σ.

Definition 3.11

A $F_{\sqsubseteq}$–environment ρ **satisfies** the basis Γ and subtyping assumptions Σ, written $\rho \models \Sigma, \Gamma$, iff for all $x : \sigma \in \Gamma$, $\rho(x) \in \Im[z]$ for some $z \sqsubseteq [\sigma]\rho$ and for all $\alpha \sqsubseteq \xi \in \Sigma$, $\rho(\alpha) \sqsubseteq [\xi]\rho$.

For example, if we have $x : real \in \Gamma$ and $\rho(x) = 5^{int}$ then ρ satisfies Γ as $5^{int} \in \Im[int]$ and $int \sqsubseteq [real]\rho = real$. Obviously we can convert 5^{int} into 5^{real} using the subtyping morphism $\Im[int \sqsubseteq real]$.

Each $F_{\sqsubseteq}$–term constant corresponds to a given element in one of the primitive types.

Definition 3.12

The function Ev assigns elements of primitive types to term constants :

$$Ev : C \to \bigcup_{d \in D} \Im[d] \text{ with } Ev(c^{\sigma}) \in \Im[\sigma].$$

At this point we are ready to define our semantics functions with the help of the functor $\Im$.

Definition 3.13

For any derivable statement $\Sigma, \Gamma \vdash M : \sigma$ and any $F_{\sqsubseteq}$–environment $\rho \models \Sigma, \Gamma$ we define $[\![\Sigma, \Gamma \vdash M : \sigma]\!]\rho$ inductively over the structure of the derivation :

a) (const)
   $[\![\Sigma, \Gamma \vdash c^{\sigma} : \sigma]\!]\rho = Ev(c^{\sigma})$
b) (var)
   $[\![\Sigma, \Gamma \vdash x : \sigma]\!]\rho = \Im[z \sqsubseteq [\sigma]\rho](\rho(x))$, where $\rho(x) \in \Im[z]$
c) $(\to E)$
   $[\![\Sigma, \Gamma \vdash MN : \tau]\!]\rho = ([\![\Sigma, \Gamma \vdash M : \sigma \to \tau]\!]\rho)_{[\sigma]\rho}\ ([\![\Sigma, \Gamma \vdash N : \sigma]\!]\rho)$
d) $(\to I)$
   $[\![\Sigma, \Gamma \vdash \lambda x : \sigma . M : \sigma \to \tau]\!]\rho$

$= \lambda d \in D_{\Im}.\text{if } d \notin [\sigma]\rho \text{ then } \lambda a \in \Im[d].() \text{ else } \lambda a \in \Im[d].[\![\Sigma,\Gamma,x:\sigma \vdash M:\tau]\!]([x \mapsto a]\rho)$

e) $(\forall E)$

$[\![\Sigma,\Gamma \vdash M\xi:\sigma[\alpha:=\xi]]\!]\rho = \Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto [\xi]\rho]\rho)]([\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]\rho)$

f) $(\forall I)$

$[\![\Sigma,\Gamma \vdash \lambda\alpha \sqsubseteq \tau.M:\forall \alpha \sqsubseteq \tau.\sigma]\!]\rho$

$=$ the unique $\kappa \in \Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho]$, such that

$\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)](\kappa) = [\![\Sigma,\alpha \sqsubseteq \tau,\Gamma \vdash M:\sigma]\!]([\alpha \mapsto d]\rho)$, for all $d \sqsubseteq [\tau]\rho$.

g) $[\![\Sigma,\Gamma \vdash M:\tau]\!]\rho = \Im[[\sigma]\rho \sqsubseteq [\tau]\rho]([\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho)$

We have to show that the definition is well formed. We do it in the following theorem.

Theorem 3.14

Let $\Im:D_{\Im} \Rightarrow CPO_{\perp}$ be a functor with property Λ (see Definition 3.20, below). For all derivable statements $\Sigma,\Gamma \vdash M:\sigma$ and for all $\rho \models \Sigma,\Gamma$, $[\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho$ is well defined and $[\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho \in \Im[[\sigma]\rho]$.

Proof

The proof goes by structural induction.

The cases (const), (var), $(\rightarrow E)$, $(\forall E)$ and (Inh) are trivial.

In case $(\rightarrow I)$ we need to show that the commutivity requirement for natural transformations holds and that the functions $\lambda a \in \Im[d].[\![\Sigma,\Gamma,x:\sigma \vdash M:\tau]\!]([x \mapsto a]\rho)$ are continuous. The first is trivial and the second is the result of Proposition 3.15.

Case $(\forall I)$ follows from Proposition 3.17.

❑

We state the needed propositions and postpone their proofs to the next section.

Proposition 3.15

Let $\Im:D_{\Im} \Rightarrow CPO_{\perp}$ be a functor with property Λ. Also let $[\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho$ be well defined for all $\rho \models \Sigma,\Gamma$. Then for every $x:\tau \in \Gamma$ and every $d \sqsubseteq [\tau]\rho$ the function $\lambda a \in \Im[d].[\![\Sigma,\Gamma \vdash M:\sigma]\!]([x \mapsto a]\rho)$ is continuous.

❑

Definition 3.16

Let $E \subseteq Env_{F_{\sqsubseteq}}$ be a set of $F_{\sqsubseteq}$–environments. We say that E is **sound** with respect to Σ,Γ iff for all $\rho \in E$, $\rho \models \Sigma,\Gamma$ holds and for every $\rho' \in E$, ρ and ρ' are equal at Γ.

Proposition 3.17

Let $\Im:D_{\Im} \Rightarrow CPO_{\perp}$ be a functor with property Λ. Furthermore let $[\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho$ be well

defined for all $\rho \in E \subseteq Env_{F_\sqsubseteq}$, where E is some set of sound environments. Then there exists a unique $\kappa \in \Im[\sqcap\{[\sigma]\rho \mid \rho \in E\}]$, st. for all $\rho \in E$: $\Im[\sqcap\{[\sigma]\rho' \mid \rho' \in E\} \sqsubseteq [\sigma]\rho](\kappa) = [\![\Sigma,\Gamma \vdash M:\sigma]\!]\rho$.

❑

The reason why we delay the proofs is that they rely on some properties of the functor $\Im$, which we define now.

Definition 3.18

Let $F:D_F \Rightarrow CPO_\perp$ be a functor in $Func_{CPO_\perp}$. The functor F **preserves directed colimits** iff for every directed set $M \subseteq D_F$, $F[\hat{\sqcup}M]$ is the colimit of $\{F[m] \mid m \in M\}$.

Definition 3.19

The functor $F:D_F \Rightarrow CPO_\perp$ is called **order faithful** iff D_F is bounded complete and for any nonempty subset $Q \subseteq D_F$ and any $a,b \in F[\sqcap Q]$:

$$(F[\sqcap Q \sqsubseteq q](a) \sqsubseteq F[\sqcap Q \sqsubseteq q](b) \text{ for all } q \in Q) \Rightarrow (a \sqsubseteq_{\sqcap Q} b).$$

Definition 3.20

A functor $F:D_F \Rightarrow CPO_\perp$ in $Func_{CPO_\perp}$ **has the property Λ** iff F preserves directed colimits, it is order faithful, and D_F is a Scott–domain (an algebraic, bounded complete cpo).

We will be able to prove Proposition 3.15 and Proposition 3.17 if the functor $\Im$ has the property Λ. Whether $\Im$ satisfies Λ depends on the functor $F_B:B \Rightarrow CPO_\perp$. In the next section we will see that if B is finite and bounded complete and F_B is order faithful, then $\Im$ will have property Λ.

We defined the meaning of a term $\Sigma,\Gamma \vdash M:\sigma$ with respect to a valid derivation of the typing judgment $M:\sigma$. Because of the rule (Inh) a typing judgment in $F_\sqsubseteq$ can have many possible derivations. Obviously we expect that the meaning of a term does not depend on the choice of the derivation for the typing judgment. We need a coherence result. (We write $[\![\Sigma,\Gamma \vdash M:\sigma]\!]_\Delta\rho$ for the meaning of a term $M:\sigma$ in the environment ρ with respect to the derivation Δ.)

Theorem 3.21

Let Δ_1 be a derivation of $\Sigma,\Gamma \vdash M:\sigma$ and let Δ_2 be a derivation of $\Sigma,\Gamma \vdash M:\tau$. Then for any $\beta \sqsupseteq \sigma,\tau$ $\Im[[\sigma]\rho \sqsubseteq [\beta]\rho]([\![\Sigma,\Gamma \vdash M:\sigma]\!]_{\Delta_1}\rho) = \Im[[\tau]\rho \sqsubseteq [\beta]\rho]([\![\Sigma,\Gamma \vdash M:\tau]\!]_{\Delta_2}\rho)$.

Proof

The proof goes by induction over the derivation of typing judgments and is rather

lengthy. Here we show the case ($\to$ I).
Let Δ_1 and Δ_2 be the following two derivations (where the last steps use the (Inh) rule only) :

$$\Delta_1 \qquad \dfrac{\dfrac{\Sigma,\Gamma,x{:}\sigma \vdash M{:}\tau_1}{\Sigma,\Gamma \vdash \lambda x{:}\sigma.M{:}\sigma\to\tau_1} \quad \Sigma \vdash \sigma\to\tau_1 \sqsubseteq \xi_{1,1}\to\zeta_{1,1}}{\vdots}}{\Sigma,\Gamma \vdash \lambda x{:}\sigma.M{:}\xi_1\to\zeta_1}$$

$$\Delta_2 \qquad \dfrac{\dfrac{\Sigma,\Gamma,x{:}\sigma \vdash M{:}\tau_2}{\Sigma,\Gamma \vdash \lambda x{:}\sigma.M{:}\sigma\to\tau_2} \quad \Sigma \vdash \sigma\to\tau_2 \sqsubseteq \xi_{2,1}\to\zeta_{2,1}}{\vdots}}{\Sigma,\Gamma \vdash \lambda x{:}\sigma.M{:}\xi_2\to\zeta_2}$$

Furthermore let $\xi_1\to\zeta_1 \sqsubseteq \xi\to\zeta$ and $\xi_2\to\zeta_2 \sqsubseteq \xi\to\zeta$.
We want to show that $\Im[\,[\xi_1\to\zeta_1]\rho \sqsubseteq [\xi\to\zeta]\rho\,]\,(\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\xi_1\to\zeta_1]\!]\rho\,) = \Im[\,[\xi_2\to\zeta_2]\rho \sqsubseteq [\xi\to\zeta]\rho\,]\,(\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\xi_2\to\zeta_2]\!]\rho\,)$.
We know that $\xi \sqsubseteq \xi_1,\xi_2 \sqsubseteq \ldots \sqsubseteq \xi_{1,1},\xi_{2,1} \sqsubseteq \sigma$ and that $\tau_1,\tau_2 \sqsubseteq \zeta_{1,1},\zeta_{2,1} \sqsubseteq \ldots \sqsubseteq \zeta_1,\zeta_2 \sqsubseteq \zeta$.

$\Im[\,[\xi_1\to\zeta_1]\rho \sqsubseteq [\xi\to\zeta]\rho\,]\,(\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\xi_1\to\zeta_1]\!]\rho\,)$
= (see page 11)
$\quad \lambda d\in D_\Im.\Im[\,[\xi_1\to\zeta_1]\rho\,(d) \sqsubseteq [\xi\to\zeta]\rho\,(d)\,] \circ (\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\xi_1\to\zeta_1]\!]\rho\,)_d$
$= \lambda d\in D_\Im.\Im[\,[\xi_1\to\zeta_1]\rho\,(d) \sqsubseteq [\xi\to\zeta]\rho\,(d)\,]$
$\qquad \circ (\Im[\,[\sigma\to\tau_1]\rho \sqsubseteq [\xi_1\to\zeta_1]\rho\,]\,(\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\sigma\to\tau_1]\!]\rho)\,)_d$
$= \lambda d\in D_\Im.\Im[\,[\xi_1\to\zeta_1]\rho\,(d) \sqsubseteq [\xi\to\zeta]\rho\,(d)\,]$
$\qquad \circ (\Im[\,[\sigma\to\tau_1]\rho\,(d) \sqsubseteq [\xi_1\to\zeta_1]\rho\,(d)\,] \circ (\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\sigma\to\tau_1]\!]\rho\,)_d\,)$
$= \lambda d\in D_\Im.\Im[\,[\sigma\to\tau_1]\rho\,(d) \sqsubseteq [\xi\to\zeta]\rho\,(d)\,] \circ (\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\sigma\to\tau_1]\!]\rho\,)_d$
$= \lambda d\in D_\Im.\Im[\,[\sigma\to\tau_1]\rho\,(d) \sqsubseteq [\xi\to\zeta]\rho(d)\,]$
$\qquad \circ (\text{ if } d\notin[\sigma]\rho \text{ then } \lambda a\in\Im[d].() \text{ else } \lambda a\in\Im[d].[\![\Sigma,\Gamma,x{:}\sigma\vdash M{:}\tau_1]\!]([x\mapsto a]\rho)\,)$
= (by induction hypothesis)
$\quad \lambda d\in D_\Im.\Im[\,[\sigma\to\tau_2]\rho(d) \sqsubseteq [\xi\to\zeta]\rho(d)\,]$
$\qquad \circ (\text{ if } d\notin[\sigma]\rho \text{ then } \lambda a\in\Im[d].() \text{ else } \lambda a\in\Im[d].[\![\Sigma,\Gamma,x{:}\sigma\vdash M{:}\tau_2]\!]([x\mapsto a]\rho)\,)$
=
$\vdots$
$= \Im[\,[\xi_2\to\zeta_2]\rho \sqsubseteq [\xi\to\zeta]\rho\,]\,(\,[\![\Sigma,\Gamma\vdash\lambda x{:}\sigma.M{:}\xi_2\to\zeta_2]\!]\rho\,)$

❑

The desired coherence result is an immediate consequence of the previous theorem.

Corollary 3.22

Let Δ_1 and Δ_2 be two derivations of the same typing judgment $M:\sigma$. Then $([\![\Sigma,\Gamma\vdash M:\sigma]\!]_{\Delta_1}\rho)=([\![\Sigma,\Gamma\vdash M:\sigma]\!]_{\Delta_2}\rho)$.

❑

At this point we are able to prove that our model is sound with respect to the equational theory of $F_{\sqsubseteq}$.

Theorem 3.23

Let $\Sigma,\Gamma\vdash M \leftrightarrow N:\sigma$ be a derivable equation in $F_{\sqsubseteq}$. Then $([\![\Sigma,\Gamma\vdash M:\sigma]\!]\rho)=([\![\Sigma,\Gamma\vdash N:\sigma]\!]\rho)$.

Proof

The proof is easy. We show the case (Eq appl2).

For the two derivations :

$$\frac{\dfrac{\Sigma,\Gamma\vdash M:\forall\alpha\sqsubseteq\sigma.\tau}{\Sigma,\Gamma\vdash M\xi_1:\tau[\alpha:=\xi_1]}}{\Sigma,\Gamma\vdash M\xi_1:\zeta} \quad\text{and}\quad \frac{\dfrac{\Sigma,\Gamma\vdash N:\forall\alpha\sqsubseteq\sigma.\tau}{\Sigma,\Gamma\vdash N\xi_2:\tau[\alpha:=\xi_2]}}{\Sigma,\Gamma\vdash N\xi_2:\zeta}$$

we get :

$[\![\,\Sigma,\Gamma\vdash M\xi_1:\zeta\,]\!]\rho$

$= \Im[\,[\tau[\alpha:=\xi_1]]\rho \sqsubseteq [\zeta]\rho\,]\,([\![\,\Sigma,\Gamma\vdash M\xi_1:\tau[\alpha:=\xi_1]\,]\!]\rho)$

$= \Im[\,[\tau[\alpha:=\xi_1]]\rho \sqsubseteq [\zeta]\rho\,]\,(\Im[\,[\forall\alpha\sqsubseteq\sigma.\tau]\rho \sqsubseteq [\tau[\alpha:=\xi_1]]\rho\,]\,([\![\,\Sigma,\Gamma\vdash M:\forall\alpha\sqsubseteq\sigma.\tau\,]\!]\rho)$

$= \Im[\,[\forall\alpha\sqsubseteq\sigma.\tau]\rho \sqsubseteq [\zeta]\rho\,]\,([\![\,\Sigma,\Gamma\vdash M:\forall\alpha\sqsubseteq\sigma.\tau\,]\!]\rho)$

= (by inductive hypothesis)

$\Im[\,[\forall\alpha\sqsubseteq\sigma.\tau]\rho \sqsubseteq [\zeta]\rho\,]\,([\![\,\Sigma,\Gamma\vdash N:\forall\alpha\sqsubseteq\sigma.\tau\,]\!]\rho)$

=

▮

$= [\![\,\Sigma,\Gamma\vdash N\xi_2:\zeta\,]\!]\rho$

❑

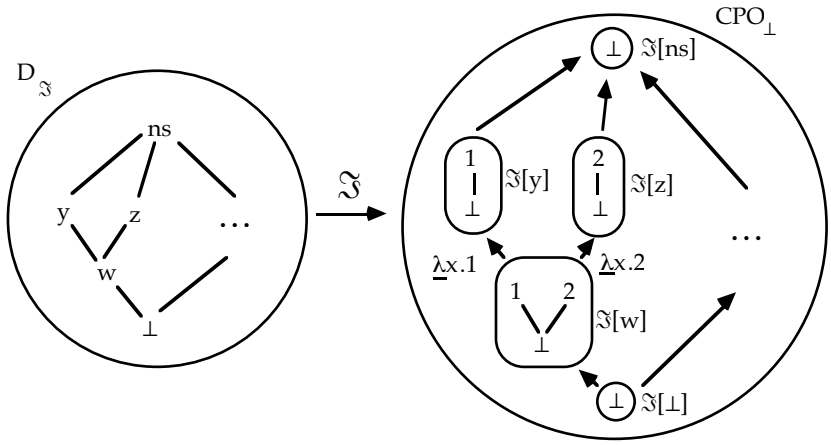
4 Theory behind the model

In this section we deliver the missing proofs from the previous section. As stated earlier, the functor $\Im$ must have the property Λ, defined in Definition 3.20. With some examples we will see what could go wrong if $\Im$ does not satisfy the Λ–property.

The following example shows what goes wrong if the functor $\Im$ is not order faithful.

Example 4.1

Let $\Im : D_\Im \Rightarrow CPO_\perp$ be a functor specified by the following figure :



The functor $\Im$ obviously is not order faithful as $w=\sqcap\{y,z\}$, $\Im[w \sqsubseteq y](1) \sqsubseteq \Im[w \sqsubseteq y](2)$ and $\Im[w \sqsubseteq z](1) \sqsubseteq \Im[w \sqsubseteq z](2)$ but $1 \not\sqsubseteq 2$ in $\Im[w]$. Now consider the following derivation :

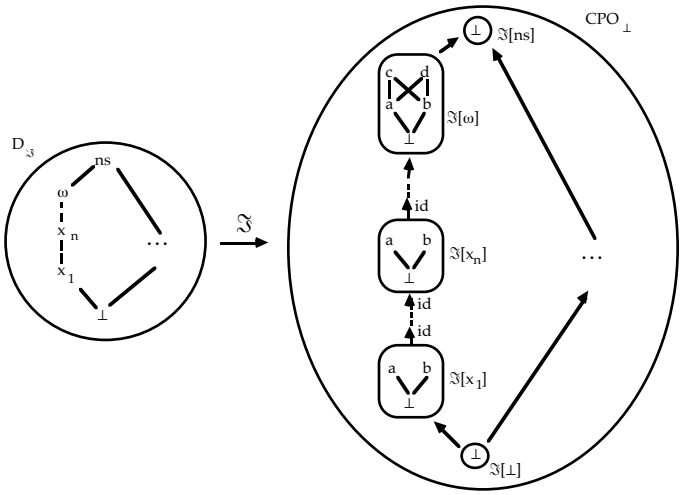
$$\frac{x : \alpha \vdash x : \alpha}{\vdash \lambda x : \alpha . x : \alpha \to \alpha}$$

Define $E := \{\rho_y, \rho_z\} \subseteq Env_{F_\sqsubseteq}$, where $\rho_y = [\alpha \mapsto y]\rho_\emptyset$ and $\rho_z = [\alpha \mapsto z]\rho_\emptyset$. E is sound with respect to the base $\Gamma = \emptyset$. In the environment ρ_y the type of $\vdash \lambda x : \alpha . x : \alpha \to \alpha$ is the function f_y, st. $f_y(y) = f_y(w) = F_y(\perp) = y$ and $f_y(q) = ns$ else. In the environment ρ_z we have the function f_z with $f_z(z) = f_z(w) = f_z(\perp) = z$ and $f_z(q) = ns$ else. When we apply the semantics definitions to the statement $\vdash \lambda x : \alpha . x : \alpha \to \alpha$, we get $[\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_y = \eta \in \Im \dashrightarrow \Im \circ f_y$ where $\eta_y = id_{\Im[y]}$, $\eta_w = \underline{\lambda} x \in \Im[w].1$, $\eta_\perp = \lambda x . \perp_{\Im[y]}$ and $\eta_q = \lambda x . ()$ for all remaining q. Similarly, $[\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_z = \gamma \in \Im \dashrightarrow \Im \circ f_z$ with $\gamma_z = id_{\Im[z]}$, $\gamma_w = \underline{\lambda} x \in \Im[w].2$, $\gamma_\perp = \lambda x . \perp_{\Im[z]}$ and $\gamma_q = \lambda x . ()$. The infimum of $f_y = [\alpha \to \alpha]\rho_y$ and $f_z = [\alpha \to \alpha]\rho_z$ is the function $f_{y \sqcap z}$ with $f_{y \sqcap z}(y) = y$, $f_{y \sqcap z}(z) = z$, $f_{y \sqcap z}(w) = f_{y \sqcap z}(\perp) = w$ and $f_{y \sqcap z}(q) = ns$ otherwise. Now we look for a unique element $\kappa \in \Im[f_{y \sqcap z}] = \Im \dashrightarrow \Im \circ f_{y \sqcap z}$, st. $\Im[f_{y \sqcap z} \sqsubseteq f_y](\kappa) = [\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_y$ and $\Im[f_{y \sqcap z} \sqsubseteq f_z](\kappa) = [\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_z$. The reader will notice that we have four different elements κ^i ($i \in \{1 \ldots 4\}$) that can be transformed back to $[\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_y$ and $[\![\vdash \lambda x : \alpha . x : \alpha \to \alpha]\!]\rho_z$. $(\kappa^i)_y = id_{\Im[y]}$ and $(\kappa^i)_z = id_{\Im[z]}$ for all i. But there are four possible functions from $\Im[w]$ to $\Im \circ f_{y \sqcap z}[w] = \Im[w]$: $(\kappa^1)_w = \underline{\lambda} x.1$, $(\kappa^2)_w = \underline{\lambda} x.2$, $(\kappa^3)_w = id_{\Im[w]}$ and $(\kappa^4)_w = [\perp \to \perp \mid 1 \mapsto 2 \mid 2 \mapsto 1]$. So the uniqueness property is violated. What is worse, from those four different (κ^i)'s we cannot choose any special element, as the set $\{\kappa^i\}$ does not have a smallest nor a biggest element. This example shows that if $\Im$ is not order faithful then Proposition 3.17 does not hold.

The next example will show what can go wrong if the functor $\Im$ does not preserve directed colimits.

Example 4.2

Again we define the functor $\Im : D_{\Im} \Rightarrow CPO_{\perp}$ with the help of the following figure :



The functor $\Im$ does not preserve directed colimits as $\omega = \bigsqcup\!\!\uparrow\{x_n\}$, but $\Im[\omega]$ is not the colimit of $\Im[x_1] \to \Im[x_2] \to \ldots \to \Im[x_n] \to \ldots$. We look one more time at the statement $\vdash \lambda x:\alpha.x:\alpha \to \alpha$. Define $E := \{\rho_n \mid n \in \mathbb{N}\} \subseteq Env_{\lambda}$ where $\rho_n = [\alpha \mapsto x_n]\rho_{\varnothing}$. Notice that $[\alpha \mapsto \omega]\rho_{\varnothing}$ is not in E. Again E is sound with respect to Γ. For all n, $[\![\vdash \lambda x:\alpha.x:\alpha \to \alpha]\!]\rho_n = \eta^n \in \Im \dashrightarrow \Im \circ (\lambda x.\text{if } x \sqsubseteq x_n \text{ then } x_n \text{ else ns})$ with $(\eta^n)_{x_k} = id_{\Im[n]} \circ \Im[x_k \sqsubseteq x_n]$ for all $x_k \sqsubseteq x_n$ and $(\eta^n)_{x_q} = \lambda x.()$ else. (In particular $(\eta^n)_{\omega} = \lambda x.()$.) We have $\bigsqcap\{[\alpha \to \alpha]\rho_n \mid n \in N\} = \lambda x.\text{if } x \sqsubseteq \omega \text{ then } x \text{ else ns}$. As in the previous example we have four different (κ^i)'s in $\Im[\bigsqcap\{[\alpha \to \alpha]\rho_n \mid n \in N\}]$, st. $\Im[\bigsqcap\{[\alpha \to \alpha]\rho_n \mid n \in N\} \sqsubseteq [\alpha \to \alpha]\rho_n](\kappa^i) = [\![\vdash \lambda x:\alpha.x:\alpha \to \alpha]\!]\rho_n$. Also there is no way to determine which one of the (κ^i)'s should be chosen. So if $\Im$ does not preserve directed colimits, Proposition 3.17 fails.

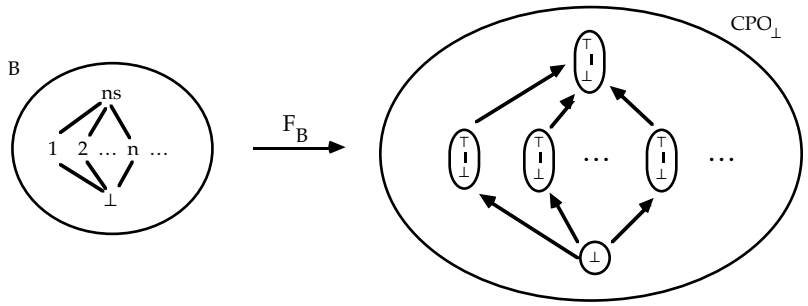
How can we guarantee that the functor $\Im$, which is the fixed point of the functor $\mathcal{K} : Func_{CPO_{\perp}} \Rightarrow Func_{CPO_{\perp}}$, will have the property Λ? It is safe to assume that all the functors in the chain :

$$\Delta = \quad \perp \xrightarrow{\perp \to \mathcal{K}[\perp]} \mathcal{K}[\perp] \xrightarrow{F[\perp \to \mathcal{K}[\perp]]} \mathcal{K}^2[\perp] \xrightarrow{\mathcal{K}^2[\perp \to \mathcal{K}[\perp]]} \cdots$$

must have the desired properties. ($\Im$ is the colimit of this chain, c.f. Definition 2.4). As we start this construction with the functor $F_B : B \Rightarrow CPO_{\perp}$ we should make sure that F_B is the right functor. But is it enough to require that F_B is order faithful and that it preserves directed colimits? Unfortunately no.

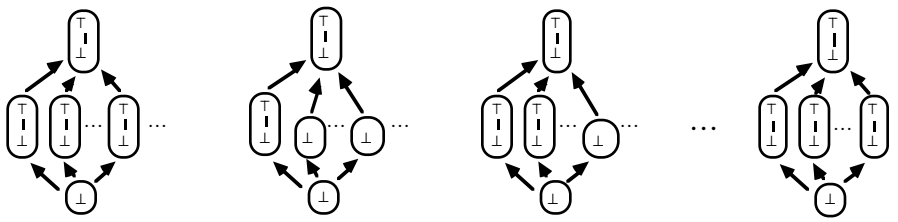
Example 4.3

The functor $\mathcal{K}$ was defined as $\mathcal{K}[F]=(F_B+(F\to F))^{ns}$. So if all $K^i[\bot]$ have property Λ then $\to : Func_{CPO_\bot}{}^{op}\times Func_{CPO_\bot}\to Func_{CPO_\bot}$ should preserve this property. But this is not the case. We define the functor $F_B : B \Rightarrow CPO_\bot$ as :



The functor $F_B\to F_B : [B\to B]\Rightarrow CPO_\bot$ with $F_B\to F_B[f]=F_B \dot{\to} F_B\circ f$ maps functions in $[B\to B]$ into cpos of natural transformations.

Next we define the functions $f_i : B\to B$ as $f_i(\bot)=\bot$, $f_i(ns)=ns$, $f_i(k)=ns$ for $k\in\{1,\dots,i\}$ and $f_i(k)=\bot$ for $k\in\{i+1,\dots\}$. Obviously $f_i\sqsubseteq f_{i+1}$ and $\hat{\bigsqcup}\{f_i\}=\underline{\lambda}x.ns$. In all the cpos $F_B \dot{\to} F_B\circ f_i$ we have only one natural transformation $\eta^\bot$ with $(\eta^\bot)_n=\lambda x.\bot$. But in $F_B \dot{\to} F_B\circ(\hat{\bigsqcup}\{f_i\})$ there are two natural transformations $\eta^\bot$ and $\eta^\top$ with $(\eta^\top)_n=id$. So $F_B\to F_B[\hat{\bigsqcup}\{f_i\}]$ is not the colimit of the cpos $F_B\to F_B[f_i]$.



The functor $\to$ also does not preserve order faithfulness.

Example 4.4

We need to modify Example 4.2 slightly. The functor $\Im$ is order faithful on the chain $\{x_n\}$, and assume that it is also order faithful for the rest of the domain $D_\Im$. Define the subset $X\subseteq D_\Im\to D_\Im$ as $X:=\{[\alpha\to\alpha]\rho_n \mid n\in N\}$. For all n and all four (κ^i)'s we have $\Im[\sqcap\{[\alpha\to\alpha]\rho_n \mid n\in N\}\sqsubseteq[\alpha\to\alpha]\rho_n](\kappa^i) = \Im[\sqcap\{[\alpha\to\alpha]\rho_n \mid n\in N\} \sqsubseteq [\alpha\to\alpha]\rho_n](\kappa^j)$ but not $\kappa^i\sqsubseteq\kappa^j$. So $\Im\to\Im$ is not order faithful.

Does this mean we cannot construct the functor $\Im$ with property Λ? Not necessarily. We

will have to modify the functor F_B slightly more. The solution is to require that the domain B of primitive types is finite. In this case all the functors in the chain :

$$\Delta = \quad \bot \xrightarrow{\bot \to \mathcal{K}[\bot]} \mathcal{K}[\bot] \xrightarrow{F[\bot \to \mathcal{K}[\bot]]} \mathcal{K}^2[\bot] \xrightarrow{\mathcal{K}^2[\bot \to \mathcal{K}[\bot]]} \cdots$$

are functors from a finite domain into $CPO_\bot$. (The function space and the sum of two finite domains are finite.) Obviously all these functors $\mathcal{K}^i[\bot]$ preserve directed colimits as all directed sets in M are finite and therefore $\bigsqcup M \in M$. Also all $\mathcal{K}^i[\bot]$ are order faithful as long as F_B is.

Lemma 4.5

Let $F:D_F \Rightarrow CPO_\bot$ be an order faithful functor in $Func_{CPO_\bot}$ with a finite domain D_F and $G:D_G \Rightarrow CPO_\bot$ be a functor with D_G finite. Then the functor $G \to F:[D_G \to D_F] \Rightarrow CPO_\bot$ is also order faithful.

Proof

As $D_G \to D_F$ is finite all subsets $X \subseteq D_G \to D_F$ must also be finite. So the infimum of X can be taken "pointwise" : $\bigsqcap X = \lambda d \in D_F.\bigsqcap\{f(d) \mid f \in X\}$. Now let $\eta, \gamma \in G \dashrightarrow F \circ (\bigsqcap X)$, st. $G \to F[(\bigsqcap X) \sqsubseteq f](\eta) \sqsubseteq G \to F[(\bigsqcap X) \sqsubseteq f](\gamma)$ for all $f \in X$. Then of course for $d \in D_G$ and $x \in G[d]$ we must have : $F[(\bigsqcap X)(d) \sqsubseteq f(d)](\eta_d(x)) \sqsubseteq F[(\bigsqcap X)(d) \sqsubseteq f(d)](\gamma_d(x))$ for all $f \in X$. So $\eta_d(x) \sqsubseteq \gamma_d(x)$ for all $x \in G[d]$ as F is order faithful. This shows that $\eta \sqsubseteq \gamma$.

❑

It is straightforward to show that G^{ns}, F+G and F×G are order–faithful functors as long as F and G are. So we know that all the functors in the chain $\bot \to \mathcal{K}[\bot] \to \ldots \to \mathcal{K}^i[\bot] \to \ldots$ have property Λ. Does the colimit of this chain, which is the λ–functor $\mathfrak{I}$, have the property Λ? First we check order faithfulness.

Proposition 4.6

Let Δ be an ω–chain $(A_n:D_{A_n} \Rightarrow CPO_\bot, (f_n:D_{A_n} \to D_{A_{n+1}}, \kappa:A_n \dashrightarrow A_{n+1} \circ f_n))$ in $Func_{CPO_\bot}{}^E$ and let $\mu:\Delta \to A$ be the colimit for Δ with $\mu_n = (h_n:D_{A_n} \to D_A, \eta_n:A_n \dashrightarrow A \circ h_n)$. If all functors A_n are order faithful then the functor A is also order faithful.

Proof

$A:D_A \Rightarrow CPO_\bot$ is the colimiting functor. Let $X \subseteq D_A$ and $a,b \in A[\bigsqcap X]$, st. $A[\bigsqcap X \sqsubseteq x](a) \sqsubseteq A[\bigsqcap X \sqsubseteq x](b)$ for all $x \in X$. μ is an O–colimit (we work in an O–category) and therefore $\bigsqcup\{A[h_n \circ (h_n)^R(\bigsqcap X) \sqsubseteq \bigsqcap X] \circ (\eta_n)_{(h_n)^R(\bigsqcap X)} \circ ((\eta_n)^R)_{\bigsqcap X}\} = id_{A[\bigsqcap X]}$. As projections preserve infima, for all n we get $(h_n)^R(\bigsqcap X) = \bigsqcap\{(h_n)^R(x) \mid x \in X\}$. Also $A_n[(h_n)^R(\bigsqcap X) \sqsubseteq (h_n)^R(x)](((\eta_n)^R)_{\bigsqcap X}(a)) \sqsubseteq A_n[(h_n)^R(\bigsqcap X) \sqsubseteq (h_n)^R(x)](((\eta_n)^R)_{\bigsqcap X}(b))$. As A_n is order

faithful we can deduce that $((\eta_n)^\kappa)_{\sqcap X}(a) \sqsubseteq ((\eta_n)^\kappa)_{\sqcap X}(b)$. From this it follows that $A[h_n \circ (h_n)^R(\sqcap X) \sqsubseteq \sqcap X] \circ (\eta_n)_{(h_n)^R(\sqcap X)} \circ ((\eta_n)^R)_{\sqcap X}(a) \sqsubseteq A[h_n \circ (h_n)^R(\sqcap X) \sqsubseteq \sqcap X] \circ (\eta_n)_{(h_n)^R(\sqcap X)} \circ ((\eta_n)^R)_{\sqcap X}(b)$. Now it is immediately clear that $a \sqsubseteq b$.

❑

We still have to show that we can construct $\Im$ in such way that it preserves directed colimits.

Proposition 4.7

Let Δ be an ω–chain $(A_n : D_{A_n} \Rightarrow CPO_\perp, (f_n : D_{A_n} \to D_{A_{n+1}}, \kappa : A_n \dashrightarrow A_{n+1} \circ f_n))$ in $Func_{CPO_\perp}{}^E$ and let $\mu : \Delta \to A$ be the colimit for Δ with $\mu_n = (h_n : D_{A_n} \to D_A, \eta_n : A_n \dashrightarrow A \circ h_n)$. If for all functors A_n the domain D_{A_n} is finite then the colimiting functor $A : D_A \Rightarrow CPO_\perp$ preserves directed colimits.

Proof

We construct the limiting source $\upsilon : A_\omega \to \Delta^R$ in $Func_{CPO_\perp}{}^P$ as in the proof to Proposition 2.5 and show that it preserves directed colimits. This functor is also colimiting for Δ [Smyth and Plotkin 1982]. Notice that $(\mu_n)^R = \upsilon_n$ and that the morphisms in Δ^R are $((f_n)^R : D_{A_{n+1}} \to D_{A_n}, (\kappa_n)^R : A_{n+1} \dashrightarrow A_n \circ (f_n)^R)$.

Let M be a directed set in D_ω with supremum $\bigsqcup\!\!\!\!\wedge M = (\bigsqcup_0, \bigsqcup_1, \ldots, \bigsqcup_n, \ldots)$. Elements $m \in M$ are of the form $(m_0, m_1, \ldots, m_n, \ldots)$ and of course for all i : $\bigsqcup_i = \bigsqcup\{m_i \mid m \in M\}$. Because all the domains D_{A_n} are finite all directed sets Q in them are finite with $\bigsqcup\!\!\!\!\wedge Q \in Q$. Therefore for all $\bigsqcup_i$ there must exist some $m \in M$ with $\bigsqcup_i = m_i$. This property will play an important role in our proof. To prove that A_ω preserves all directed colimits, we must verify the colimit conditions for all directed sets M and $\bigsqcup\!\!\!\!\wedge M$ [Fiech 1996]. We prove here only the condition $(X = X_F{}^*)$. The proof of the remaining conditions can be found in [Fiech 1993]

Condition $(X = X_F{}^*)$

We will show that for every element $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots) \in A_\omega[\bigsqcup\!\!\!\!\wedge M]$ there exists a directed subset $Q \subseteq \bigcup\{A_\omega[m \sqsubseteq \bigsqcup\!\!\!\!\wedge M](\delta) \mid \delta \in A[m] \text{ for some } m \in M\}$, st. $\alpha = \bigsqcup\!\!\!\!\wedge Q$. How can we construct this set? There must be some element $m \in M$ with $m_0 = \bigsqcup_0$. In this $A_\omega[m]$ we can construct the first "approximation" β to α. We start with $\beta_0 = \alpha_0 \in A_0[m_0]$ and construct $\beta_{n+1} \in A_{n+1}[m_{n+1}]$ from β_n as follows : $\beta_{n+1} = A_{n+1}[f_n(m_n) \sqsubseteq m_{n+1}] \circ (\kappa_n)_{m_n}(\beta_n)$. Obviously $\beta_{n+1} \in A_{n+1}[m_{n+1}]$ and $((\kappa_n)^R)_{m_{n+1}}(\beta_{n+1}) = \beta_n$. So we have the element $\beta = (\beta_0, \beta_1, \ldots, \beta_n, \ldots) \in A_\omega[m]$. Now we must show that $A_\omega[m \sqsubseteq \bigsqcup\!\!\!\!\wedge M](\beta) \sqsubseteq \alpha$. It is enough to show that $(A_n[m_n \sqsubseteq \bigsqcup_n](\beta_n) \sqsubseteq \alpha_n) \Rightarrow (A_{n+1}[m_{n+1} \sqsubseteq \bigsqcup_{n+1}](\beta_{n+1}) \sqsubseteq \alpha_{n+1})$ as we start with $\beta_0 = \alpha_0$. We have $A_{n+1}[m_{n+1} \sqsubseteq \bigsqcup_{n+1}](\beta_{n+1}) = A_{n+1}[m_{n+1} \sqsubseteq \bigsqcup_{n+1}] \circ A_{n+1}[f_n(m_n) \sqsubseteq m_{n+1}] \circ (\kappa_n)_{m_n}(\beta_n) = A_{n+1}[f_n(m_n) \sqsubseteq \bigsqcup_{n+1}] \circ (\kappa_n)_{m_n}(\beta_n) = A_{n+1}[f_n(\bigsqcup_n) \sqsubseteq \bigsqcup_{n+1}] \circ A_{n+1}[f_n(m_n) \sqsubseteq f_n(\bigsqcup_n)] \circ (\kappa_n)_{m_n}(\beta_n) = A_{n+1}[f_n(\bigsqcup_n) \sqsubseteq \bigsqcup_{n+1}] \circ$

$(\kappa_n)_{\sqcup_n} \circ A_n[m_n \sqsubseteq \sqcup_n](\beta_n) \sqsubseteq A_{n+1}[f_n(\sqcup_n) \sqsubseteq \sqcup_{n+1}] \circ (\kappa_n)_{\sqcup_n}(\alpha_n) \sqsubseteq \alpha_{n+1}$. In this way for every n we can construct an element $\beta^n \in A_\omega[m^n]$, st. $(m^n)_i = \sqcup_i$ and $(\beta^n)_i = \alpha_i$ for $i \le n$. Obviously $A_\omega[m^n \sqsubseteq \hat{\sqcup} M](\beta^n) \sqsubseteq A_\omega[m^{n+1} \sqsubseteq \hat{\sqcup} M](\beta^{n+1})$ and $\alpha = \hat{\sqcup}\{A_\omega[m^n \sqsubseteq \hat{\sqcup} M](\beta^n) \mid n \in N\}$. So the first condition is satisfied.

❑

We can prove now that the functor $\Im$ with property Λ exists.

Theorem 4.8

Let $F_B : B \Rightarrow CPO_\perp$ be an order–faithful functor with a finite domain B. Then the functor $\Im$ from Definition 2.4 has the property Λ.

Proof

The theorem follows immediately from Proposition 4.7, Proposition 4.6 and Lemma 4.5. $D_\Im$ is a Scott–domain as it is the colimit of finite (therefore algebraic) and bounded complete domains.

❑

One nice property of a functor $F : D_F \Rightarrow CPO_\perp$ with property Λ is that natural transformations are uniquely determined through functions on finite elements in D_F, $K(D_F)$.

Lemma 4.9

Let $F : D_F \Rightarrow CPO_\perp$ be a functor with property Λ and $F' : D_F \Rightarrow CPO_\perp$ any other functor in $Func_{CPO_\perp}$ with the same domain D_F. If $H := \{h_a : F[a] \to F'[a] \mid a \in K(D_F)\}$ is a set of functions, st. the corresponding diagrams commute: $F'[a \sqsubseteq b] \circ h_a = h_b \circ F[a \sqsubseteq b]$, then there exists a unique natural transformation $\eta : F \dot{\to} F'$ with $\eta_a = h_a$ for all $a \in K(D_F)$.

Proof

We have to define η_ω for $\omega \notin K(D_F)$. D_F is Scott–domain and therefore $\omega = \hat{\sqcup}\{d \mid d \sqsubseteq \omega$ and $d \in K(D_F)\}$. As F preserves directed colimits, $F[\omega]$ is the colimit of $\{F[d] \mid d \in (\downarrow\omega) \cap K(D_F)\}$. Define the natural sink $(F[d] \to F'[\omega])_{d \in K(D_F)}$ with $f_d = F'[d \sqsubseteq \omega] \circ h_d$. Now η_ω is the unique function from $F[\omega]$ to $F'[\omega]$ that makes the diagrams commute. It is clear that η is a natural transformation.

❑

Now that we know that we can work with the functor $\Im$ with property Λ we are ready to prove the two propositions from section 4.

Proof 4.10 (Proposition 3.15)

The proof proceeds by structural induction. We show here the cases ($\rightarrow$E) and ($\forall$I). The remaining cases can be found in [Fiech 1993].

($\rightarrow$E)

Assume that $\lambda a \in \Im[d].[\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto a]\rho)$ and $\lambda a \in \Im[d].[\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto a]\rho)$ are continuous functions. Let now $x:\sigma \in \Gamma$ and let $M \subseteq \Im[d]$ be a directed set with $d \sqsubseteq [\sigma]\rho$. We have to show that

$[\![\Sigma,\Gamma \vdash MN:\tau]\!]([x \mapsto \hat{\bigsqcup} M]\rho) = \hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash MN:\tau]\!]([x \mapsto m]\rho) \mid m \in M\}$.

$[\![\Sigma,\Gamma \vdash MN:\tau]\!]([x \mapsto \hat{\bigsqcup} M]\rho) =$

$([\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto \hat{\bigsqcup} M]\rho))_{[\sigma]\rho}([\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto \hat{\bigsqcup} M]\rho)) =$

$(\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto m]\rho) \mid m \in M\})_{[\sigma]\rho}(\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto m^*]\rho) \mid m^* \in M\}) =$

$(\hat{\bigsqcup}\{([\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto m]\rho))_{[\sigma]\rho} \mid m \in M\})(\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto m^*]\rho) \mid m^* \in M\}) =$

$\hat{\bigsqcup}\{([\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto m]\rho))_{[\sigma]\rho}(\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto m^*]\rho) \mid m^* \in M\}) \mid m \in M\} =$

$\hat{\bigsqcup}\{\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto m]\rho))_{[\sigma]\rho}(\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto m]\rho)) \mid m^* \in M\} \mid m \in M\} =$

$\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash M:\sigma \rightarrow \tau]\!]([x \mapsto m]\rho))_{[\sigma]\rho}([\![\Sigma,\Gamma \vdash N:\sigma]\!]([x \mapsto m]\rho)) \mid m \in M\} =$

$\hat{\bigsqcup}\{[\![\Sigma,\Gamma \vdash MN:\tau]\!]([x \mapsto m]\rho) \mid m \in M\}$.

($\forall$I)

From the definition of $[\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]\rho$ we get for all $d \sqsubseteq [\tau]\rho$:

$\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)]([\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto \hat{\bigsqcup} M]\rho)) =$

$[\![\Sigma,\alpha \sqsubseteq \tau,\Gamma \vdash M:\sigma]\!]([\alpha \mapsto d, x \mapsto \hat{\bigsqcup} M]\rho) =$

$\hat{\bigsqcup}\{[\![\Sigma,\alpha \sqsubseteq \tau,\Gamma \vdash M:\sigma]\!]([\alpha \mapsto d, x \mapsto m]\rho) \mid m \in M\}$.

So for all $d \sqsubseteq [\tau]\rho$ and $m \in M$

$\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)]([\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto m]\rho)) =$

$[\![\Sigma,\alpha \sqsubseteq \tau,\Gamma \vdash M:\sigma]\!]([\alpha \mapsto d, x \mapsto m]\rho) \sqsubseteq$

$\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)]([\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto \hat{\bigsqcup} M]\rho))$.

The functor $\Im$ is order faithful and $[\forall \alpha \sqsubseteq \tau.\sigma]\rho = \bigsqcap\{[\sigma]([\alpha \mapsto d]\rho \mid d \sqsubseteq [\tau]\rho\}$.

Therefore $[\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto m]\rho) \sqsubseteq [\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto \hat{\bigsqcup} M]\rho)$ for all $m \in M$.

For any other upper bound κ of the set $\{[\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto m]\rho) \mid m \in M\}$ we get that for all $d \sqsubseteq [\tau]\rho$ $\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)](\kappa)$ is the upper bound of $\{[\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([\alpha \mapsto d, x \mapsto m]\rho) \mid m \in M\}$ and hence $\Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)]$ $([\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto m]\rho)) \sqsubseteq \Im[[\forall \alpha \sqsubseteq \tau.\sigma]\rho \sqsubseteq [\sigma]([\alpha \mapsto d]\rho)](\kappa)$. Using again the property that $\Im$ is order faithful we get $[\![\Sigma,\Gamma \vdash M:\forall \alpha \sqsubseteq \tau.\sigma]\!]([x \mapsto \hat{\bigsqcup} M]\rho) \sqsubseteq \kappa$.

❑

Proof 4.11 (Proposition 3.17)

We use again structural induction. We need to show only the existence of κ as uniqueness follows from the fact that $\mathfrak{I}$ is order faithful. Here we show cases $(\to I)$ and $(\forall E)$ and refer the reader to [Fiech 1993] for the remaining cases.

$(\to I)$

We are looking for the unique element κ in $\mathfrak{I}[\sqcap\{[\sigma\to\tau]\rho \mid \rho\in E\}]$.

For finite elements $b\in K(D_{\mathfrak{I}})$, $(\sqcap\{[\sigma\to\tau]\rho \mid \rho\in E\})(b) = \sqcap\{([\sigma\to\tau]\rho)(b) \mid \rho\in E\}$. Define $E_b := \{\rho\in E \mid b\sqsubseteq[\sigma]\rho\}$. Then $(\sqcap\{[\sigma\to\tau]\rho \mid \rho\in E\})(b) = \sqcap\{[\tau]\rho \mid \rho\in E_b\}$. For b with $E_b=\emptyset$ we have $\kappa_b:\mathfrak{I}[b]\to\mathfrak{I}[ns]$ with $\kappa_b=\lambda x.()$. For every $b\in K(D_{\mathfrak{I}})$ with $E_b\neq\emptyset$ we define $\kappa_b:\mathfrak{I}[b]\to\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}]$ as $\kappa_b := \lambda a\in\mathfrak{I}[b].\alpha$, where α is the unique element in $\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}]$, st. $\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}\sqsubseteq[\tau]\rho](\alpha) = [\Sigma,\Gamma,x:\sigma\vdash M:\tau]([x\mapsto a]\rho)$ for all $\rho\in E_b$. (Notice that the set $\{[x\mapsto a]\rho \mid \rho\in E_b\}$ is sound with respect to Σ,Γ as $b\sqsubseteq[\sigma]\rho$ for all $\rho\in E_b$.)

That κ_b is continuous can be shown as in case $(\forall I)$ on page 30. We need to prove now that the diagrams commute. For all $\rho\in E_c$ and $b\sqsubseteq c\in D_{\mathfrak{I}}$:

$\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_c\}\sqsubseteq[\tau]\rho]\circ(\kappa_c\circ\mathfrak{I}[b\sqsubseteq c]) =$

$(\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_c\}\sqsubseteq[\tau]\rho]\circ\kappa_c)\circ\mathfrak{I}[b\sqsubseteq c] =$

$id_{\mathfrak{I}[[\tau]\rho]}\circ(\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}\sqsubseteq[\tau]\rho]\circ\kappa_b) =$

$\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_c\}\sqsubseteq[\tau]\rho]\circ(\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}\sqsubseteq\sqcap\{[\tau]\rho \mid \rho\in E_c\}]\circ\kappa_b)$.

From this we get that $\kappa_c\circ\mathfrak{I}[b\sqsubseteq c] = \mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E_b\}\sqsubseteq\sqcap\{[\tau]\rho \mid \rho\in E_c\}]\circ\kappa_b$ as $\mathfrak{I}$ is order faithful. So the diagrams commute for all $a,b\in K(D_{\mathfrak{I}})$.

The set of functions $\{\kappa_a:\mathfrak{I}[a]\to\mathfrak{I}[\sqcap\{[\tau]\rho \mid \rho\in E\} \mid a\in K(D_{\mathfrak{I}})]$ can be completed in a unique way to a natural transformation $\kappa\in\mathfrak{I}\dot{\to}\mathfrak{I}\circ(\sqcap\{[\sigma\to\tau]\rho \mid \rho\in E\})$ (Lemma 4.9). It is clear that $\mathfrak{I}[\sqcap\{\sigma\to\tau]\rho \mid \rho\in E\}\sqsubseteq[\sigma\to\tau]\rho](\kappa) = [\![\Sigma,\Gamma\vdash\lambda x.M:\sigma\to\tau]\!]\rho$ for all $\rho\in E$.

$(\forall E)$

There exists the unique element $\gamma\in\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}$, st.

$\mathfrak{I}[\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}\sqsubseteq[\forall\alpha\sqsubseteq\tau.\sigma]\rho](\gamma) = [\![\Sigma,\Gamma\vdash M:\forall\alpha\sqsubseteq\tau.\sigma]\!]\rho$ for all $\rho\in E$.

We claim that $\mathfrak{I}[\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}\sqsubseteq\sqcap\{[\sigma[\alpha:=\tau]]\rho \mid \rho\in E\}](\gamma) =: \kappa$ is the necessary element in $\sqcap\{([\sigma[\alpha:=\tau]]\rho \mid \rho\in E\}$.

$\mathfrak{I}[\sqcap\{[\sigma[\alpha:=\tau]]\rho \mid \rho\in E\}\sqsubseteq[\sigma[\alpha:=\tau]]\rho](\kappa) =$

$\mathfrak{I}[\sqcap\{[\sigma[\alpha:=\tau]]\rho \mid \rho\in E\}\sqsubseteq[\sigma[\alpha:=\tau]]\rho]$

$\quad\circ\,\mathfrak{I}[\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}\sqsubseteq\sqcap\{[\sigma[\alpha:=\tau]]\rho \mid \rho\in E\}](\gamma) =$

$\mathfrak{I}[\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}\sqsubseteq[\sigma[\alpha:=\tau]]\rho](\gamma) =$

$\mathfrak{I}[[\forall\alpha\sqsubseteq\tau.\sigma]\rho\sqsubseteq[\sigma[\alpha:=\tau]]\rho]\circ\mathfrak{I}[\sqcap\{[\forall\alpha\sqsubseteq\tau.\sigma]\rho \mid \rho\in E\}\sqsubseteq[\forall\alpha\sqsubseteq\tau.\sigma]\rho](\gamma) =$

$\mathfrak{I}[[\forall\alpha\sqsubseteq\tau.\sigma]\rho\sqsubseteq[\sigma]([\alpha\mapsto[\tau]\rho]\rho)]\ ([\Sigma,\Gamma\vdash M:\forall\alpha\sqsubseteq\tau.\sigma]\rho) =$

$[\![\Sigma,\Gamma\vdash M:\sigma[\alpha:=\tau]]\!]\rho$ for all $\rho\in E$.

❑

5 Extensions of $F_{\sqsubseteq}$

It is straightforward to include records and record types in our model.
The set of untyped λ–terms Λ is defined now as

$$\Lambda = V \mid C \mid \Lambda\Lambda \mid \lambda V:T.\Lambda \mid \Lambda.l$$

The type names of the calculus are defined by the following grammar

$$\mathbb{T} = \mathbb{V} \mid \mathbb{C} \mid \mathbb{T}\to\mathbb{T} \mid \forall\mathbb{V}\sqsubseteq\mathbb{T}.\mathbb{T} \mid \{l_1:\mathbb{T}_1,\ldots,l_n:\mathbb{T}_n\}$$

We have the standard subtyping rule for record types

$$\text{(Sub rec)}\quad \frac{\Sigma\vdash\sigma_1\sqsubseteq\tau_1,\ldots,\ \Sigma\vdash\sigma_n\sqsubseteq\tau_n}{\Sigma\vdash\{l_1:\sigma_1,\ldots l_n:\sigma_n,l_{n+1}:\sigma_{n+1},\ldots,l_m:\sigma_m\}\sqsubseteq\{l_1:\tau_1,\ldots,l_n:\tau_n\}}$$

The type assignment rules for records and field selection are as follows :

$$\text{(Rec)}\quad \frac{\Sigma,\Gamma\vdash M_1:\sigma_1,\ldots,\ \Sigma,\Gamma\vdash M_n:\sigma_n}{\Sigma,\ \Gamma\vdash\{l_1:M_1,\ldots,l_n:M_n\}:\{l_1:\sigma_1,\ldots,l_n:\sigma_n\}}$$

$$\text{(Sel)}\quad \frac{\Sigma,\Gamma\vdash M:\{l_1:\sigma_1,\ldots,l_n:\sigma_n\}}{\Sigma,\Gamma\vdash M.l_i:\sigma_i}\qquad 1\le i\le n$$

The function $[\ _\ ]:\mathbb{T}\times Env_{type}\to D_{\Im}$ is extended to record type names :

$$[\ \{l_1:\sigma_1,\ldots,l_n:\sigma_n\}\ ]\rho = \underline{\lambda}l\in L.\ \text{if } l=l_i \text{ then } [\sigma_i]\rho \text{ else } ns$$

We generalize the definition of $D_{\Im}$ to $D_{\Im}\cong(B+(D_{\Im}\to D_{\Im})+(L\to|T|))^{ns}$, where L is the flat cpo of record labels. The functor $\Im : D_{\Im}\Rightarrow CPO_{\perp}$, is now the fixed point of $\mathcal{K}:Func_{CPO_{\perp}}\Rightarrow Func_{CPO_{\perp}}$ with $\mathcal{K}[F]=(F_B+(F\to F)+(\perp_L\to F))^{ns}$. We extend it to functions $g\in L\to D_{\Im}$:

$\Im[g\in L\to|T|] = \perp_L \dashrightarrow \Im\circ g$
$\Im[g\sqsubseteq g'] = \lambda\gamma\in\perp_L \dashrightarrow \Im\circ g.(\lambda l\in L.\ \Im[g(d)\sqsubseteq g'(d)]\circ\gamma_l)$
(Here, $\perp_L : L\Rightarrow CPO_{\perp}$, is the constant functor with $\perp_L[l]=\{\perp\}$ for all $l\in L$.)

If L is a finite cpo, then the functor $\Im$ has the property Λ (the requirements on F_B remain the same).

The additional semantics definitions are :

$$[\![\Sigma,\Gamma\vdash\{l_1:M_1,\ldots,l_n:M_n\}:\{l_1:\sigma_1,\ldots,l_n:\sigma_n\}]\!]\rho$$

$$= \lambda l \in L.\text{if } l=l_i \text{ then } \lambda a \in \{\bot\}.[\![\Sigma,\Gamma \vdash M_i : \sigma_i]\!]\rho \text{ else } \lambda a \in \{\bot\}.()$$

$$[\![\Sigma,\Gamma \vdash M.l_i : \sigma_i]\!]\rho = ([\![\Sigma,\Gamma \vdash M : \{l_1 : \sigma_1,\ldots,l_n : \sigma_n\}]\!]\rho)_{l_i}(\bot)$$

As indicated in the introduction, the type $[\forall \alpha \sqsubseteq \{l:\text{int}\}.\alpha \to \alpha\,]\rho$ has more elements than just the approximations of the identity natural transformations. This allows us to define constants for polymorphic record update. For example we can define the constant update_l_to_5 : $\forall \alpha \sqsubseteq \{l:\text{real}\}.\alpha \to \alpha$, which given a record with the field l of type real (resp. int) updates the field to the value 5.0 resp. 5. The function $[\forall \alpha \sqsubseteq \{l:\text{real}\}]\rho$ is the restricted identity function $\text{id}\,|_{d \sqsubseteq [\{l:\text{real}\}]\rho}$ ($d \not\sqsubseteq [\{l:\text{real}\}]\rho$ is mapped to ns).

We can extend the model in a straightforward way to intersection types, and it appears relatively easy to extend the model to F_ω (some details still need to be checked). We also believe that we will be able to apply our model to recursive types. At this point we can provide interpretation only to inductive types ($\mu t.\alpha$, where t occurs positively in α [Amadio 1991]). We can also enforce that all types are interpreted as Scott–domains and to do so we need all the cpos that interprete basic types be finite. (In general the cpo of natural transformations $F \dot{\to} F \circ f$ is not algebraic – for necessary conditions consult [Fiech, Huth 1994]).

6 Bibliography

Abadi M. and Plotkin G.D. [1990]
A Per Model of Polymorphism and Recursive Types, Proc. Symposium on Logic in Computer Science, IEEE 1990, pp. 355-365.

Abadi M. [1994]
Baby Modula-3 and theory of objects, Journal of Functional Programming 4,2 pp.249-283

Adamek J., Herrlich H., Strecker G. [1990]
Abstract and Concrete Categories, John Wiley&Sohns, 1990

Amadio R.M. [1991]
Recursion over Realizability Structures, Information and Computation 91,1 pp.55-86

Amadio R.M., Bruce K., Longo G. [1986]
The finitary projection model for second order lambda calculus and solutions to higher order domain equations, in Logic in Computer Science, IEEE 1986, pp.122-130

Barendregt H.P. and Hemerik K.[1990]
Types in Lambda Calculi and Programming Languages, in: "ESOP'90, 3rd Symposium on Programming", Copenhagen, May 15–18, 1990, LNCS, vol.432

Barendregt H.P. [1984]
The Lambda Calculus: Its Syntax and Semantics, Studies in Logic and the Foundations of Mathematics, vol.103, North–Holland, Amsterdam 1984

Breazu-Tannen V., Coquand T., Gunter C.A., Scedrov A. [1989]
Inheritance and Explicit Coersion, Symposium on Logic in Computer Science, IEEE 1989, pp.112-129

Breazu-Tannen V., Coquand T., Gunter C.A., Scedrov A. [1991]
Inheritance as Implicit Coersion, Information and Computation 93,1 pp.172-221

Bruce K., Longo G. [1990]
A Modest Model of Records, Inheritance and Bounded Quantification, Information and Computation vol. 87, pp. 196-240

Bruce K., Meyer A., Mitchell J. [1990]
The Semantics of Second–Order Lambda–Calculus, Information and Computation vol. 85, pp. 76-134

Bruce K., Mitchell J. [1992]
PER models of subtyping, recursive types and higher–order polymorphism, Proc. ACM Symposium on Principles of Programming Languages 1992, pp.316-327

Cardelli L., Martini S., Mitchell J., Scedrov A. [1991]
An extension of system F with subtyping, International Conference on Theoretical Aspects of Computer Software, LNCS, vol. 526, pp.750-770.

Cardelli L., Wegner P. [1985]
On Understanding Types, Data Abstraction, and Polymorphism, ACM Computing Surveys 17,4 pp.471-522

Cardone F. [1989]
Relational Semantics for Recusive Types and Bounded Quantification, ICALP 1989, in LNCS 372, pp.164-178.

Cardone F. [1991]
Recursive types for Fun, Theoretical Computer Science 83, pp.29-56

Coppo M. [1985]
A completeness theorem for recursively defined types, Proc. 12th International Colloquium on Automata, Languages and Programming, LNCS vol.194 pp.120-129.

Coquand T., Gunter C.A., Winskel G. [1987]
DI–domains as a model of polymorphism, in Mathematical Foundations of Programming Language Semantics, LNCS vol. 298 pp. 344-363

Coquand T., Gunter C.A., Winskel G. [1989]
Domain theoretic models of polymorphism, Information and Computation 81, 2 pp. 123-167

Fiech A. [1993]
A Denotational Model for Polymorphic Lambda Calculus with Subtyping, Ph.D. thesis, Kansas State University, Manhattan, USA

Fiech A. [1996]
Colimits in the Category DCPO, Mathematical Structures in Computer Science, vol. 6, pp. 1–14

Fiech A., Huth M. [1994]
Algebraic domains of natural transformations, Theoretical Computer Science 136, pp. 57–78

Gunter C., Mitchell J. (edt.) [1994]
Theoretical Aspects of Object–Oriented Programming, The MIT Press, 1994

Girard J.–Y. [1972]
Interpretation Fonctionnelle et Elimination des Coupures dans l'Arithmetique d'Ordre Superieur, Ph.D.–thesis, Universite Paris VII (1972)

MacQueen D., Plotkin G., Sethi R. [1986]
An ideal model for recursive polymorphic types, Information and Control 71, pp. 95-130

Meyer A.R. [1982]
What is a model of the lambda calculus, Information and Control 52, pp. 87-122

Reynolds J.C. [1974]
Towards a Theory of Type Structure, Proceedings, Colloque sur la Programmation, LNCS 19, pp.408-425

Reynolds J.C. [1980]
Using Category Theory to Design Implicit Conversions and Generic Operators, in LNCS 94, p. 211-258, Springer Verlag 1980

Reynolds J.C. [1983]
Types, abstraction and parametric polymorphism, Information Processing 83, pp. 513-523, Elsevier Science Publishers B.V., Amsterdam

Schmidt D. [1990]
Action Semantics Based Language Design, SOFSEM '90 Zotavovna Sirena, Janske Lazne, Krkonose, Nov. 1990.

Seely, R.A.G [1987]
Categorical Semantics for Higher Order Polymorphic Lambda Calculus, Journal of Symbolic Logic 52, pp.969-989

Smyth M.B. and Plotkin, G. [1982]
The Category–Theoretic Solution of Recursive Domain Equations. SIAM J.Computing 11 (1982) 761–783.