A Full Continuous Model of Polymorphism

Franco Barbanera[1] and Stefano Berardi[2]

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Catania,
Viale A. Doria 6, 95125 Catania (Italy). barba@dmi.unict.it
[2] Dipartimento di Informatica, Universitá degli studi di Torino,
Corso Svizzera 185, 10149 Torino (Italy). stefano@di.unito.it

Abstract. We introduce a model of the second-order lambda calculus. Such a model is a Scott domain whose elements are themselves Scott domains, and in it polymorphic maps are interpreted by generic continous maps.

Keywords: Second-order lambda calculus, model, Scott domain, non-parametric.

1 Introduction

In this paper we define the Full model, a model of the second-order lambda calculus (λ_2). In the Full model, polymorphic maps are interpreted by *generic* continuous maps, that is, maps really depending on input types.

Some readers might argue that the interesting models of system λ_2 are the parametric ones, where only constant or "almost" constant polymorphic maps are considered. These models have been often used in the literature for many different purposes, but would not be of help for the implicit goal of the present paper, that is to provide a semantic basis for extensions of system λ_2 where one can define computations really depending on the "type tag" of their input.

It is not difficult to imagine programming languages where both functional and imperative features are present, and where it could be possible, and useful, to define polymorphic computations really depending on the "type tag" of their input.

Suppose, for instance, to have an extension of λ_2 containing the traditional atomic types $\mathtt{Int}, \mathtt{Char}, \mathtt{Bool}, \mathtt{Real}$, added to improve efficiency. We could have also the following primitive polymorphic command, $\mathtt{ToString} : \forall\alpha.(\alpha \to \mathtt{String})$, taking any type α, any $a : \alpha$, and "printing" it (returning a string out of it). Typically, this map would be defined by case, calling a specific printing procedure for $\mathtt{Int}$, another one for $\mathtt{Char}, \mathtt{Bool}, \mathtt{Real}, \ldots$, and printing a warning message whenever one tries to print an element of a function type. $\mathtt{ToString}$ is an *essentially non-constant* (hence non-parametric) polymorphic map; the same is true for polymorphic order tests, polymorphic sorting maps, and so on. More involved examples would arise if we mixed classes from object-oriented languages with second order lambda calculus. Indeed, in object oriented languages, the application of a function to an argument may produce different results according to the type of the input.

The intuition underlying the Full model . As a matter of fact our non-parametric model of λ_2 is not the first model in the literature which allows to model polymorphic maps really depending on input types [1], [3], [6], [8]. However, we claim our Full model has a simpler definition.

The Full model consists of two Scott domains, $\mathtt{Types}$ and $\mathtt{Terms}$. $\mathtt{Types}$ represents the types of λ_2, and $\mathtt{Terms}$ the terms of λ_2. Each $X \in \mathtt{Types}$ (each ”type”) is itself a Scott domain, and a subdomain of $\mathtt{Terms}$. The elements $x \in X$ will in turn interpret terms of λ_2 having type X. Both terms and types are obtained as ”consistent” sets of atoms. We have two notions of ”consistency” on atoms, one used to build terms, which we call ”coherence”, and another one used to build types, which we call ”homogeneity”. Two atoms are coherent if they may be two pieces of the same datum; they are homogeneous if they are pieces of data having the same type. Say, the atoms 0 and 1 are not coherent, because no integer datum can be, at the same time, both 0 and 1. On the other hand, 0 and 1 are homogeneous, because they are both data of type $\mathtt{Int}$.

The model is obtained using an Engeler model construction twice, once to define the Scott domain $\mathtt{Types}$, the other to define the Scott domain $\mathtt{Terms}$. Some extra conditions are needed in order for terms and types to match within the model. Interpretation of second order features of λ_2 then works as one would expect. Type constructors of λ_2 are interpreted as continous maps $F : \mathtt{Types} \rightarrow \mathtt{Types}$. ”Polymorphic maps” associated to such an F are interpreted as continuous maps $f : \mathtt{Types} \rightarrow \mathtt{Terms}$ such that $f(\alpha) \in F(\alpha)$ for all $\alpha \in \mathtt{Types}$. Quantification over F is interpreted by a type $\forall\alpha.F(\alpha) \in \mathtt{Types}$, whose elements are exactly *all* polymorphic maps associated to F.

Our Full model includes, as we shall see through examples in section 4, non-constant maps defined by cases over types. Again by an example, we shall show that it does not satisfy axiom C (a weaker form of parametricity). Hence the Full model is provably not parametric.

Beta-Eta completeness The Full model has also an unexpected and nice theoretical feature: it equates two terms of λ_2 if and only if such terms are $\beta\eta$-convertible. In other words, the Full model is $\beta\eta$-complete. The proof generalizes Friedman $\beta\eta$-completeness proof of set-theoretical model of first order lambda calculus and may be found in [5].

The paper is organised as follows. In Section 2 we recall the definition of the second-order polymorphic lambda-calculus and of what is a model for it. Section 3 is devoted to the costruction of our Full model. In the conclusion (Section 4) we present and discuss some relevant features of the Full model.

All the proofs of the paper, but the proof of the correctness of the Full model, will be given in the Appendix A.

2 The calculus and its models

In this section, mostly in order to fix the notation, we quickly recall the definition of the second-order polymorphic lambda-calculus (λ_2) and of what is a model for it.

The **types** of λ_2 are formed according to the following grammar

$$\sigma ::= \mathsf{C} \mid \mathsf{t} \mid \sigma \rightarrow \sigma \mid \forall \mathsf{t}.\sigma$$

where C ranges over a set of *Type Constants* and t ranges over a set of *Type Variables*.

The **terms** of λ_2 are formed according to the following grammar

$$M ::= \mathsf{c} \mid \mathsf{x} \mid \lambda \mathsf{x} : \sigma.M \mid \lambda \mathsf{t}.M \mid (MM) \mid (M\sigma)$$

where c ranges over a set of *Term Constants* and x ranges over a set of *Term Variables*.

By defining *contexts* as sets of the form $\Gamma = \{x_1 : \sigma_1, \ldots, x_n : \sigma_n\}$, the **typing rules** of λ_2 can be presented as follows

$$(var)\ \Gamma \rhd x : \sigma \quad (x : \sigma \in \Gamma)$$

$$(cst)\ \Gamma \rhd c : \sigma \quad (c \text{ a constant of type } \sigma)$$

$$(\rightarrow I)\ \frac{\Gamma, x : \tau \rhd M : \sigma}{\Gamma \rhd \lambda x : \tau.M : \tau \rightarrow \sigma}$$

$$(\rightarrow E)\ \frac{\Gamma \rhd M : \tau \rightarrow \sigma\ \Gamma \rhd N : \tau}{\Gamma \rhd MN : \sigma}$$

$$(\forall I)\ \frac{\Gamma \rhd M : \sigma}{\Gamma \rhd \lambda t.M : \forall t.\sigma} \quad (t \text{ not free in } \Gamma)$$

$$(\forall E)\ \frac{\Gamma \rhd M : \forall t.\sigma}{\Gamma \rhd M\tau : \sigma[\tau/t]}$$

Two **notions of reduction** are defined on terms of λ_2.

β-reduction: $(\lambda x : \tau.M)N \rightarrow M[N/x]$

type-β-reduction: $(\lambda t.M)\tau \rightarrow M[\tau/t]$

We refer to the standard references, for instance [13], for the definition of the reduction relation induced by the two notions of reduction above, for the definition of term- and type-substitution in λ_2 and for all usual notations and conventions.

We recall now two definitions, of structure and of model for λ_2, as presented in [7] (see also [13]). A λ_2-applicative structure, or a structure for λ_2, is a structure in which the connectives of system λ_2 are interpreted by some operation in the model. This spells out as follows.

Definition 2.1 (λ_2-applicative structures). *A* λ_2-applicative structure $\mathcal{A}$ *is a tuple*

$$\mathcal{A} = \langle \mathcal{U}, Dom, \{\mathbf{App}^{a,b}, \mathbf{App}^{f}\}, \mathcal{I}\rangle$$

where

- $\mathcal{U} = \{T^{\mathcal{A}}, [T^{\mathcal{A}} \to T^{\mathcal{A}}], \to^{\mathcal{A}}, \forall^{\mathcal{A}}, \mathcal{I}_{\mathsf{C}}\}$ *specifies a set* $T^{\mathcal{A}}$ *(the "types" of the structure), a set* $[T^{\mathcal{A}} \to T^{\mathcal{A}}]$ *of functions from* $T^{\mathcal{A}}$ *to* $T^{\mathcal{A}}$*, a binary operation* $\to^{\mathcal{A}}$ *on* $T^{\mathcal{A}}$*, a map* $\forall^{\mathcal{A}}$ *from* $[T^{\mathcal{A}} \to T^{\mathcal{A}}]$ *to* $T^{\mathcal{A}}$*, and a map* $\mathcal{I}_{\mathsf{C}}$ *from type constants to* $T^{\mathcal{A}}$*.*
- $Dom = \{Dom^a \mid a \in T^{\mathcal{A}}\}$ *is a collection of sets indexed by the types of the structure.*
- $\{\mathbf{App}^{a,b}, \mathbf{App}^{f}\}$ *is a collection of application maps, with one* $\mathbf{App}^{a,b}$ *for every pair of types* $a, b \in T^{\mathcal{A}}$ *and one* $\mathbf{App}^{f}$ *for every function* $f \in [T_{\mathcal{A}} \to T^{\mathcal{A}}]$*.*
  Each $\mathbf{App}^{a,b}$ *must be a function*

  $$\mathbf{App}^{a,b} : Dom^{a \to b} \to (Dom^a \to Dom^b)$$

  and each $\mathbf{App}^{f}$ *must be a function*

  $$\mathbf{App}^{f} : Dom^{\forall^{\mathcal{A}} f} \to \prod_{a \in T^{\mathcal{A}}} Dom^{f(a)}$$

- $\mathcal{I} : \text{Constants} \to \bigcup_{a \in T^{\mathcal{A}}} Dom^a$ *assigns a value to each constant symbol, with* $\mathcal{I}(c) \in Dom^{[\![\tau]\!]}$ *if* c *is a constant of type* τ*.* $[\![\tau]\!]$ *is the meaning of* τ *as defined below.*

A λ_2-applicative structure is *extensional* if every $\mathbf{App}^{a,b}$ and $\mathbf{App}^{f}$ is one-to-one. A structure is a Henkin model, or simply a model, if the interpretation of the connectives of λ_2 is compatible with the reductions of λ_2. Unfortunately, this simple idea requires some effort in order to be precisely formalized.

Definition 2.2 (Henkin models). *An extensional λ_2-applicative structure $\mathcal{A}$ is a* Henkin model *if, for every term* $\Gamma \triangleright M : \sigma$ *and every* $\eta \models \Gamma$*,* $[\![\Gamma \triangleright M : \sigma]\!]_{\eta}$*, as defined below, exists.*

- *An $\mathcal{A}$-environment is a mapping*

  $$\eta : \text{Variables} \to (T^{\mathcal{A}} \cup \bigcup_{a \in T^{\mathcal{A}}} Dom^a)$$

  such that for every type variable t *and term variable* x*, we have* $\eta(t) \in T^{\mathcal{A}}$ *and* $\eta(x) \in \bigcup_{a \in T^{\mathcal{A}}} Dom^a$*. We shall denote by* η^p_v *the mapping such that* $\eta^p_v(w) = \eta(w)$ *for any* $w \neq v$*, and* $\eta^p_v(v) = p$

– *The meaning* $[\![\sigma]\!]_\eta$ *of a type expression* σ *in environment* η *is defined inductively as follows*
  - $[\![t]\!]_\eta = \eta(t)$ *(t type variable)*
  - $[\![C]\!]_\eta = \mathcal{I}_{\mathsf{C}}(C)$ *(C type constant)*
  - $[\![\tau \to \tau']\!] = [\![\tau]\!]_\eta \to^{\mathcal{A}} [\![\tau']\!]_\eta$
  - $[\![\forall t.\sigma]\!]_\eta = \forall^{\mathcal{A}}(\boldsymbol{\lambda} a \in T^{\mathcal{A}}.[\![\sigma]\!]_{\eta_t^a})$.

– *If* Γ *is a context, then* η satisfies Γ, *written* $\eta \models \Gamma$, *if* $\eta(x) \in Dom^{[\![\sigma]\!]_\eta}$ *for every* $x : \sigma \in \Gamma$.

– *The meaning of a term* $\Gamma \rhd M : \sigma$ *in environment* $\eta \models \Gamma$ *is defined by induction as follows:*
  - $[\![\Gamma \rhd x : \sigma]\!]_\eta = \eta(x)$
  - $[\![\Gamma \rhd MN : \tau]\!]_\eta = \mathbf{App}^{a,b}[\![\Gamma \rhd M : \tau \to \tau']\!]_\eta[\![\Gamma \rhd N : \tau']\!]_\eta$
    where $a = [\![\tau]\!]_\eta$ *and* $b = [\![\tau']\!]_\eta$
  - $[\![\Gamma \rhd \lambda x : \sigma.M : \sigma \to \tau]\!]_\eta =$ *the unique* $f \in Dom^{a\to b}$ *s.t., for all* $d \in Dom^a$, $\mathbf{App}^{a,b} f d =$ $[\![\Gamma, x : \sigma \rhd M : \tau]\!]_{\eta_x^d}$
    where $a = [\![\sigma]\!]_\eta$ *and* $b = [\![\tau]\!]_\eta$
  - $[\![\Gamma \rhd M\tau : \sigma[\tau/t]]\!]_\eta = \mathbf{App}^f[\![\Gamma \rhd M : \forall t.\sigma]\!]_\eta[\![\tau]\!]_\eta$,
    where $f(a) = [\![\sigma]\!]_{\eta_t^a}$ *for all* $a \in T^{\mathcal{A}}$
  - $[\![\Gamma \rhd \lambda t.M : \forall t.\sigma]\!]_\eta =$ *the unique* $g \in Dom^{\forall^{\mathcal{A}} f}$ *s.t., for all* $a \in T^{\mathcal{A}}$,
    $\mathbf{App}^f g a = [\![\Gamma \rhd M : \sigma]\!]_{\eta_t^a}$
    where $f(a) = [\![\sigma]\!]_{\eta_t^a}$ *for all* $a \in T^{\mathcal{A}}$.

3 The Full model

We suppose the reader to be familiar with Engeler construction of a model of untyped lambda calculus [2]. As we anticipated in the introduction, we will repeat Engeler construction twice, one to define a Scott Domain `Terms` to interpret terms of λ_2, and the other to define a Scott Domain `Types` to interpret types of λ_2. Some extra conditions will be required to express relationships between `Terms` and `Types`. The construction will pass through three steps: the definition of a set of atoms, with a constructor for so-called "step-functions", the definition of a consistency notion on atoms, and the definition of an entailment relation between atoms.

First step: the definition of the set $\boldsymbol{\Omega}$ of atoms. We introduce a set $\boldsymbol{\Omega}$ of atoms. Terms and types of λ_2 will be interpreted as subsets of $\boldsymbol{\Omega}$ satisfying a consistency condition: coherence in the case of terms, homogeneity in the case of types.

We suppose fixed a family $\{L_i\}_i$ of disjoint sets of atomic data. These could be, for instance, $L_0 = \{0, 1, 2, \ldots\}$ (integers), $L_1 = \{\mathtt{true}, \mathtt{false}\}$ (booleans), $L_2 = \{a, b, c, \ldots\}$ (characters), $etc. \ldots$.

$\boldsymbol{\Omega}$ is defined starting from $\{L_i\}_i$ and then closing under two constructors,

$$(-,-), \langle -,- \rangle : \mathcal{P}_{fin}(\boldsymbol{\Omega}) \times \boldsymbol{\Omega} \to \boldsymbol{\Omega}$$

The constructor $(-,-)$ will denote all step functions from the domain `Terms` of terms in the model, to `Terms` itself. As usual, a step-function denoted (a,x) will map any $b \in$ `Terms` (any consistent set b of atoms) including a into the singleton $\{x\}$, and anything else into $\emptyset$ (taken to represent an "indefinite" output). Each first order function of λ_2 will be built as a pointwise union of step functions, and identified with the corresponding set of atoms. Let us consider an example by assuming the integers to be among the atomic data.

The atom $(\{n\},n)$ represents the step function mapping any element containing n into $\{n\}$ itself, and undefined elsewhere.

For any set X of atoms, the set of atoms $\mathtt{id}_X = \{(\{x\},x)|x \in X\}$, representing the pointwise union of all step functions $(\{x\},x)$, will be the identity on X.

In a similar way, the constructor $\langle -,- \rangle$ will denote all step functions from the domain `Types` of terms in the model, to `Types` (or to `Terms` itself). Each type constructor, and each polymorphic function of λ_2 will be built as pointwise union of step functions, and identified with the corresponding set of atoms. Continuing the example above, for any n the atom $\langle \{n\}, (\{n\},n) \rangle$ represents the step function mapping any type including n (say, the type of integer) into the (singleton of the) step function $(\{n\},n)$, and undefined elsewhere. The set of atoms $\mathtt{id} = \{\langle \{x\}, (\{x\},x) \rangle | x \text{ atom}\}$, representing the pointwise union of all step functions $\langle \{x\}, (\{x\},x) \rangle$, will be the polymophic identity. In fact, it will send any type X into $\mathtt{id}_X = \{(\{x\},x)|x \in X\}$ i.e, into the identity on X.

Definition 3.1 (The set $\boldsymbol{\Omega}$). *The set $\boldsymbol{\Omega}$ is the smallest set satisfying:*

1. *$L_i \subset \boldsymbol{\Omega}$, for each i;*
2. *($a \subset \boldsymbol{\Omega}$, a finite, $x \in \boldsymbol{\Omega}$) $\Rightarrow (a,x), \langle a,x \rangle \in \boldsymbol{\Omega}$.*

Second step: the definition of the consistency notion on atoms. In the construction of our Full model we shall use only a particular subset of $\boldsymbol{\Omega}$. Such a subset *Cons* will be defined together with two binary "consistency" relations on $\boldsymbol{\Omega}$: *homogeneity* (`ho`), and *coherence* (`co`). *Cons* will consists of the elements of $\boldsymbol{\Omega}$ which are both homogeneous and consistent with themselves.

A set will be said to be *homogeneous* (*coherent*) if all of its elements are pairwise *homogeneous* (*coherent*). As a matter of fact the notion of pairwise homogeneity (coherence) may have different interpretation. We leave it unspecified for the time being; it will be formally defined later on.

As we said, coherent sets will form a Scott domain `Terms` interpreting terms; homogeneous sets will form a Scott domain `Types` interpreting types, and will be themselves equipped with a structure of Scott domain.

A set a will be said to be *homogeneous* (*coherent*) with a set b, $a \,\mathtt{ho}\, b$($a \,\mathtt{co}\, b$) for short, whenever $a \cup b$ is *homogeneous* (*coherent*).

As in the Engeler construction, the choice of the clauses for $\mathtt{co}$, $\mathtt{ho}$ will be sometimes forced in order to have a model, and will be sometimes arbitrary (depending on which notion of type and polymorphic map we want to end up with). We first express (a possible choice of) conditions on $\mathtt{co}$, $\mathtt{ho}$ by words, then we will translate them into an inductive definition.

- We ask that each L_i be a flat domain of data. This means that each data type L_i will be an homogeneous set, but two different atoms in L_i will never be coherent, because they will represent pairwise incompatible values for the same datum (say, $0, 1 \in L_0$, or $\mathtt{true}, \mathtt{false} \in L_1$).
- We ask that $(a, x), (b, y)$ be coherent (two pieces of the same function) if they map coherent inputs (pieces of the same input element) into coherent outputs (pieces of the same output element). $(a, x), (b, y)$ are homogeneous (pieces of the same function type) if a, b are pieces of the same input type, and x, y are pieces of the same output type.
- We ask that $\langle a, x\rangle, \langle b, y\rangle$ be coherent (two pieces of the same type polymorphic map) if they map homogeneous inputs (pieces of the same input type) into coherent outputs (pieces of the same output element). $\langle a, x\rangle, \langle b, y\rangle$ are homogeneous (two pieces of the same polymorphic map) if they map homogeneous inputs (pieces of the same input type) into homogeneous outputs (pieces of the same output type).

In the informal definition above, we have implicitly assumed that two coherent or homogeneous elements are either both in some L_i, or both of the form (a, x), or both of the form $\langle a, x\rangle$; that is, a type may contain only data, or only first order functions, or only polymorphic functions.

If the reader takes now some time to formalize the choices of conditions expressed above, (s)he will end up with the following definition.

Definition 3.2. *(*Cons, $\mathtt{ho}$, $\mathtt{co}$*) We define the set* Cons $\subseteq \boldsymbol{\Omega}$ *and the relations* $\mathtt{ho}, \mathtt{co} \subseteq$ Cons $\times$ Cons *by simultaneous induction as follows:*

($\mathtt{ho}_0$) $\quad L_i \times L_i \subseteq \mathtt{ho}$ *for any* i*;*
($\mathtt{co}_0$) $\quad p \,\mathtt{co}\, p$ *for any* $p \in L_i$ *and any* i*;*
(Cons$_0$) $\quad L_i \subseteq$ Cons *for any* i*;*

($\mathtt{ho}_1$) $\quad (a, x) \,\mathtt{ho}\, (b, y)$ *if* $(a, x), (b, y) \in$ Cons, $a \,\mathtt{ho}\, b$ *and* $x \,\mathtt{ho}\, y$*;*
($\mathtt{co}_1$) $\quad (a, x) \,\mathtt{co}\, (b, y)$ *if* $(a, x), (b, y) \in$ Cons *and* $[a \,\mathtt{co}\, b \Rightarrow x \,\mathtt{co}\, y]$
(Cons$_1$) $\quad (a, x) \in$ Cons *if* $x \in$ Cons *and* a *is a coherent and homogeneous subset of* Cons

($\mathtt{ho}_2$) $\quad \langle a, x\rangle \,\mathtt{ho}\, \langle b, y\rangle$ *if* $\langle a, x\rangle, \langle b, y\rangle \in$ Cons *and* $[a \,\mathtt{ho}\, b \Rightarrow x \,\mathtt{ho}\, y]$*;*
($\mathtt{co}_2$) $\quad \langle a, x\rangle \,\mathtt{co}\, \langle b, y\rangle$ *if* $\langle a, x\rangle, \langle b, y\rangle \in$ Cons *and* $[a \,\mathtt{ho}\, b \Rightarrow x \,\mathtt{co}\, y]$
(Cons$_2$) $\quad \langle a, x\rangle \in$ Cons *if* $x \in$ Cons *and* a *is a homogeneous subset of* Cons

Notice that *Cons* is neither homogeneous nor coherent. In fact it contains, for instance, the two non-homogeneous elements 0 and $(\{0\},0)$, and the two non-coherent elements $(\{0\},0)$ and $(\{0\},1)$.

Remark 3.3. It is straightforward to see that the following holds:
- Any subset of an homogeneous (coherent) set is homogeneous (coherent).
- Any two subsets of an homogeneous (coherent) set are homogeneus (coherent) with each other.

We shall denote by *Cons*$|_{()}$ and *Cons*$|_{\langle\rangle}$ the subsets of *Cons* whose elements are all of the form (a,x) and $\langle a,x\rangle$.

Third (and last) step: the definition of entailments on ***Cons*****.** We introduce two entailment relations on *Cons*: $\vdash_{\texttt{co}}$ and $\vdash_{\texttt{ho}}$. Such relations are needed in order to get an extensional model of λ_2

The intuitive meaning of $a \vdash_{\texttt{co}} x$ is: x denotes a map smaller than a, or, equivalently, a and $a \cup \{x\}$ represent the same function $: \texttt{Terms} \to \texttt{Terms}$. We will check that the set $a^{\vdash_{\texttt{co}}}$, of all x such that $a \vdash_{\texttt{co}} x$, is the maximum set representing the same function as a. By bounding ourselves to subsets of *Cons* of the form $a^{\vdash_{\texttt{co}}}$, we will have just one denotation for each function. Thus, two subsets associated to the same function $: \texttt{Terms} \to \texttt{Terms}$ will be equal, and we will get an extensional model of λ_2 (extensional on terms). In the same way, $a \vdash_{\texttt{ho}} x$ intuitively means: a and $a \cup \{x\}$ represent the same function $: \texttt{Types} \to \texttt{Types}$ (or $: \texttt{Types} \to \texttt{Terms}$). By bounding ourselves to subsets of *Cons* closed under $\vdash_{\texttt{ho}}$, we will get an extensional model of λ_2 (extensional on polymorphic maps).

Definition 3.4. *($\vdash_{\texttt{ho}}, \vdash_{\texttt{co}}$)*

(i) *The relations* $\vdash_{\texttt{ho}}, \vdash_{\texttt{co}} \subseteq \mathrm{Cons} \times \mathrm{Cons}$ *are defined by simultaneous induction as follows. Let* $x,y \in \mathrm{Cons}$, $a,b \in \mathcal{P}_{fin}(\mathrm{Cons})$. *For any* $X,Y \subseteq \mathrm{Cons}$, *let* $X \vdash_{\texttt{ho}} Y (X \vdash_{\texttt{co}} Y)$ *be short for* $\forall y \in Y \exists x \in X.\, x \vdash_{\texttt{ho}} y (x \vdash_{\texttt{co}} y)$.

$$x \vdash_{\texttt{ho}} x \quad \textit{for any } x \in (\bigcup_i L_i \cup \mathrm{Cons}|_{()})$$

$$x \vdash_{\texttt{co}} x \quad \textit{for any } x \in \bigcup_i L_i$$

$$\frac{a \vdash_{\texttt{ho}} b \quad x \vdash_{\texttt{ho}} y}{\langle b,x\rangle \vdash_{\texttt{ho}} \langle a,y\rangle}$$

$$\frac{a \vdash_{\texttt{co}} b \quad x \vdash_{\texttt{co}} y}{(b,x) \vdash_{\texttt{co}} (a,y)}$$

$$\frac{a \vdash_{\mathtt{ho}} b \quad x \vdash_{\mathtt{co}} y}{\langle b, x\rangle \vdash_{\mathtt{co}} \langle a, y\rangle}$$

where $a, b \subset$ Cons, $x, y \in$ Cons *and* $a \vdash_{\mathtt{ho}} b (a \vdash_{\mathtt{co}} b)$ *is short for* $\forall y \in b \exists x \in a. x \vdash_{\mathtt{ho}} y (x \vdash_{\mathtt{co}} y)$.

(ii) *Let* $a, X \subseteq$ Cons, *then*

$$\begin{aligned} a^{\vdash_{\mathtt{ho}}} &=_{Def} \{x \in \text{Cons} \mid a \vdash_{\mathtt{ho}} x\} \\ a^{\vdash_{\mathtt{co}}} &=_{Def} \{x \in \text{Cons} \mid a \vdash_{\mathtt{co}} x\} \\ a^{X} &=_{Def} a^{\vdash_{\mathtt{co}}} \cap X. \end{aligned}$$

where $a \vdash_{\mathtt{ho}} x (a \vdash_{\mathtt{co}} x)$ *is short for* $a \vdash_{\mathtt{ho}} \{x\} (a \vdash_{\mathtt{co}} \{x\})$

Given $X \subseteq$ *Cons* we shall denote by $\mathcal{P}_{|\mathtt{ho}}(X)$ and $\mathcal{P}_{|\mathtt{co}}(X)$ the sets of, respectively, homogeneus and coherent subsets of X. The superscript "*fin*" will denote the extra restriction to *finite* subsets of X.

We are now ready to define the Scott domain interpreting types of λ_2 as the set of homogeneous subsets of *Cons* closed with respect to $\vdash_{\mathtt{ho}}$. The Scott domain interpreting terms will be instead defined as the set of coherent subsets of *Cons* closed with respect to $\vdash_{\mathtt{co}}$.

Definition 3.5.

$$\mathtt{Types} =_{Def} \{a^{\vdash_{\mathtt{ho}}} \mid a \in \mathcal{P}_{|\mathtt{ho}}(\text{Cons})\}.$$

$$\mathtt{Terms} =_{Def} \{a^{\vdash_{\mathtt{co}}} \mid a \in \mathcal{P}_{|\mathtt{co}}(\text{Cons})\}.$$

Proposition 3.6. *(i)* $(\mathtt{Types}, \subseteq, \bigcup)$ *is a Scott domain, with* $\{a_0^{\vdash_{\mathtt{ho}}} \mid a_0 \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\text{Cons})\}$ *as the set of its compact elements.*

(ii) $(\mathtt{Terms}, \subseteq, \bigcup)$ *is a Scott domain, with* $\{a_0^{\vdash_{\mathtt{co}}} \mid a_0 \in \mathcal{P}^{fin}_{|\mathtt{co}}(\text{Cons})\}$ *as the set of its compact elements.*

As usual, given a domain D, $[D \to D]$ denotes the set of the *continuous* functions from D to D. We may now introduce operation on Types interpreting arrow and quantification over types of λ_2.

Definition 3.7. *($\Rrightarrow$, Q)*

We define $\Rrightarrow$: $\mathtt{Types} \times \mathtt{Types} \to \mathtt{Types}$ *and* $\mathsf{Q} : [\mathtt{Types} \to \mathtt{Types}] \to \mathtt{Types}$ *as follows.*
Let $X, Y \in \mathtt{Types}$ *and let* $F \in [\mathtt{Types} \to \mathtt{Types}]$.

$$\begin{aligned} X \Rrightarrow Y &=_{Def} \{(a, y) \mid a \in \mathcal{P}^{fin}_{|\mathtt{co}}(X),\ y \in Y\}. \\ \mathsf{Q}(F) &=_{Def} \{\langle a, y\rangle \mid a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\text{Cons}),\ y \in F(a^{\vdash_{\mathtt{ho}}})\}. \end{aligned}$$

In the Appendix $\Rrightarrow$ and Q will be proved to be well-defined and continuous (Proposition A.2). It is possible to associate a Scott domain to any element of Types, in such a way that

$X \Rrightarrow Y$ and $\mathsf{Q}(F)$ will be the set of continuous maps from X to Y, and of "polymorphic maps associated to F" (the maps $f : \mathtt{Types} \rightarrow \mathtt{Terms}$ such that $f(X) \in F(X)$ for all $X \in \mathtt{Types}$).

The Scott domain associated to X consists of all traces to X of elements of $\mathtt{Terms}$ closed under $\vdash_{\mathsf{co}}$.

Definition 3.8. *Let* $X \in \mathtt{Types}$.

$$\| X \| =_{Def} \{a^X \mid a \in \mathtt{Terms}\}.$$

The closure under $\vdash_{\mathsf{co}}$ in $\| X \|$ is required in order to have extensionality of the interpretation (on terms). Remark that the "elements" of $\| X \|$ are not the atoms of X, but the *sets of atoms of* X (coherent and closed under $\vdash_{\mathsf{co}}$ in X).

Proposition 3.9. *For any* $X \in \mathtt{Types}$, $(\| X \|, \subseteq, \bigcup)$ *is a Scott domain, with* $\{a_0^X \mid a_0 \in \mathcal{P}^{fin}_{|\mathsf{co}}(X)\}$ *as the set of its compact elements.*

Definition 3.10. *Let* $F \in [\mathtt{Types} \rightarrow \mathtt{Types}]$. *We define*

$$[\prod_{T \in \mathtt{Types}} \| F(T) \|] =_{Def} \{f \in [\mathtt{Types} \rightarrow \mathtt{Terms}] \mid f(X) \in \| F(X) \| \text{ for } X \in \mathtt{Types}\}.$$

We consider the elements of $[\prod_{T \in \mathtt{Types}} \| F(T) \|]$ as ordered by pointwise inclusion.

It is now possible to prove that $\| X \Rrightarrow Y \|$ and $\| \mathsf{Q}(F) \|$ are isomorphic, respectively, to $[\| X \| \rightarrow \| Y \|]$ and $[\prod_{T \in \mathtt{Types}} \| F(T) \|]$. This means that we interpret our arrow and universally quantified types with as rich a set of functions as possible. It will be routine to show that what we have is indeed a model for λ_2.

Proposition 3.11. *Let* $X, Y \in \mathtt{Types}$ *and* $F \in [\mathtt{Types} \rightarrow \mathtt{Types}]$. *Then*

(i) There exists an isomorphism pair $((-)^{\Uparrow}, (-)^{\Downarrow})$ *such that*

$$\| X \Rrightarrow Y \| \simeq [\| X \| \rightarrow \| Y \|].$$

(ii) There exists an isomorphism pair $((-)^{\Uparrow}, (-)^{\Downarrow})$ *such that*

$$\| \mathsf{Q}(F) \| \simeq [\prod_{T \in \mathtt{Types}} \| F(T) \|].$$

We can now define a λ_2-applicative structure as follows. For simplicity sake we assume to have one basic type "o" and no term constants.

- $\mathcal{U} = \{\mathtt{Types}, [\mathtt{Types} \rightarrow \mathtt{Types}], \Rrightarrow, \mathsf{Q}, \mathcal{I}_{\mathsf{C}}\}$
- $Dom^X = \| X \|$ for $X \in \mathtt{Types}$
- $\mathbf{App}^{a,b} = \boldsymbol{\lambda} h \in \| a \Rrightarrow b \| . \boldsymbol{\lambda} x \in \| a \| . h^{\Uparrow}(x)$
- $\mathbf{App}^{f} = \boldsymbol{\lambda} k \in \| \mathsf{Q}(f) \| . \boldsymbol{\lambda} x \in \mathtt{Types}. k^{\Uparrow}(x)$

– $\mathcal{I}_{\mathsf{C}}(o) = I\!N$

It is easy to check that the one above is a well-defined, extensional λ_2-applicative structure. Now we can show that what we have is indeed a Henkin Model.

Theorem 3.12 (Main Theorem).
The λ_2-applicative structure above defined is a Henkin Model.

Proof. We have to show that for every term $\Gamma \triangleright M : \sigma$ and every $\eta \models \Gamma$, there exists $[\![\Gamma \triangleright M : \sigma]\!]_\eta$, as defined in Definition 2.2.
In order to do that we can prove a stronger statement by induction, namely that for every $x : \tau \in \Gamma$ and $\eta \models \Gamma$, the map

$$d \in [\![\tau]\!]_\eta \mapsto [\![\Gamma \triangleright M : \sigma]\!]_{\eta^d_x}$$

is a continuous function from $[\![\tau]\!]_\eta$ to $[\![\sigma]\!]_\eta$. By Proposition 3.11, our interpretations of the arrow types and of the universally quantified types consist of all the continuous functions of the appropriate functionality. Then the inductive proof can be easily carried on almost in the same way as the standard proof that the full continuous hierarchy is a model for the simply typed lambda calculus (see [13] for a good presentation). Of course we first need to show that for every type σ, $[\![\sigma]\!]_\eta$ exists. This result can easily be achieved by showing that for every type variable t, the map

$$X \in \texttt{Types} \mapsto [\![\sigma]\!]_{\eta^X_t}$$

is a continuous function from `Types` to `Types`, and this can be done by means of a straightforward induction on the structure of σ.

4 Comparison with a PER model

To conclude the paper we show some elementary properties of the Full model(including the fact that it is *not* parametric), and some examples of non-constant polymorphic maps. We shall also state (without proving it) the $\beta\eta$-completeness property. Such property makes clear the differences between our Full model and parametric models, for example Longo's $PER(P(\omega))$, the Partial Equivalence Relation model over the lambda model $P(\omega)$ [7]. We shall also briefly discuss about the interpretation of integers in our model.

Proposition 4.1. *(i) There is a continuous map $\mathsf{Q}' \in [\texttt{Types} \to \texttt{Types} \to \texttt{Types}]$ inverting the quantifier map Q, that is, such that $\mathsf{Q}'(\mathsf{Q}(F), X) = F(X)$.*
(ii) There is a continuous map $\mathsf{P}_{1,2} \in [\texttt{Types} \to \texttt{Types}\times\texttt{Types}]$ inverting the arrow constructor for non-empty domains, that is, such that $\mathsf{P}_{1,2}(\Rrightarrow (X, Y)) = (X, Y)$ whenever Y is not empty (it is associated to a non-empty set of atoms).

Proposition 4.2. *The Full model is not parametric. In fact it does not satisfies the weaker "axiom C" of [11].*

We shall recall the "axiom C" in the proof of the above proposition in the Appendix.

Proposition 4.3. *(i) There is a map* $\mathsf{j} : [\mathtt{Types} \to \mathtt{Types} \to \mathtt{Terms} \to \mathtt{Terms}]$*, such that* $\mathsf{j}(X, Y, x) = y \in Y$ *whenever* $x \in X$*, and* $\mathsf{j}(X, X, x) = x$ *(type recasting is the identity when* $X = Y$*).*

(ii) If L_0 *is the set of integers, and the sets* L_i *are pairwise disjoint, then there exists an element* $\mathtt{test} \in [\mathtt{Types} \to L_0]$ *which, given any* $X \in \mathtt{Types}$*, checks whether* X *is a type of first order functions, a type of polymorphic functions, or a subtype of some* L_i*.*

From a theoretical viewpoint, the most interesting (and unexpected) property of the Full model is the $\beta\eta$-completeness.

Theorem 4.4. *The Full model is* $\beta\eta$*-complete, that is the following hold*

1. *Two closed types denote the same element of* $\mathtt{Types}$ *if and only if they are* α*-convertible; and*
2. *Two closed terms of* λ_2 *denote the same element of* $\mathtt{Terms}$ *if and only if they are* $\beta\eta$*-convertible.*

We do not include the proof of the theorem in this paper: it may be found in [5]. We will rather use $\beta\eta$-completeness to point out the difference between the Full model and the $PER(P(\omega))$ model ([7]), which is parametric.

Comparing the Full model and $PER(P(\omega))$. Let $N = \forall\alpha.(\alpha \to \alpha) \to (\alpha \to \alpha)$ be the version of Church integers defined within λ_2. There exist closed terms $f, g : N \to N$ of λ_2 which are non-convertible, yet equal in the model $PER(P(\omega))$. It is enough to take f, g extensionally equal ($f(n) =_{\beta\eta} g(n)$ for all closed normal $n : N$), yet not convertible: say $f = S_l$ (the left successor), $g = S_r$ (the right successor) [1]. Then S_l, S_r are equal in $PER(P(\omega))$, but different (by $\beta\eta$-completeness) in the Full model. The reason is that, in the Full model, N is not the "right" interpretation of integers. Indeed, in the Full model, $N = \forall\alpha.(\alpha \to \alpha) \to (\alpha \to \alpha)$ consists of all polymorphic functionals sending a map over α into a map over the same α. If we have non- constant polymorphic maps, functionals in N are far more than just Church integers. For some of such extra functionals, $S_l, S_r : N \to N$ will produce two different results. Thus S_l, S_r are different in the Full model.

[1] Define $S_l = \lambda n : N.\lambda\alpha.\lambda f : (\alpha \to \alpha).\lambda x : \alpha.n(\alpha, f, f(x))$ and $S_r = \lambda n : N.\lambda\alpha.\lambda f : (\alpha \to \alpha).\lambda x : \alpha.f(n(\alpha, f, x))$. We have $S_l(n, \alpha, f, x) = f^n(f(x)) = f^{(1+n)}(x)$, while $S_l(n, \alpha, f, x) = f(f^n(x)) = f^{(n+1)}(x)$. S_l and S_r are extensionally equal over terms representing integers. In $PER(P(\omega))$ every element in the interpretation of N is equal to some integer, and the model is extensional. It follows that S_l, S_r are equal in $PER(P(\omega))$.

Interpreting integers in the Full model. One may think that a “good” model of λ_2 should equate S_l, S_r, and, thefore, that our Full model is not a ”good” model. As a matter of fact also the Full model does equate S_l, S_r, but we need to choose the “right” interpretation of integers. In the Full model, such “right” interpretation of integers is not the interpretation of N, but the flat domain $L_0 = \{0, 1, 2, \ldots\}$. Then we could add to λ_2 some fresh constants $\mathtt{Int}, 0, 1, +, *, \ldots$ denoting L_0 and some primitive operations over it. In the Full model, we have (as expected) extensional equality over terms of type $\mathtt{Int} \rightarrow \mathtt{Int}$, not just $\beta\eta$-convertibility. For instance, take any map s_l, s_r, corresponding to the left and right successor, but over the type $\mathtt{Int}$. We could define $s_l = (\lambda x : \mathtt{Int}.1 + x), s_r = (\lambda x : \mathtt{Int}.x + 1)$. s_l, s_r are equated in the Full model (we can check that they have the same trace). In fact, the type $\mathtt{Int} \rightarrow \mathtt{Int}$ is not in the original λ_2, thus the $\beta\eta$-completeness result does not apply to it. Completeness of Full model only applies to “pure” typed lambda terms, not to lambda terms containg extra constants like $\mathtt{Int}, +, \ldots$.

5 Conclusions

It has been known since the very beginning that types in a polymorphic lambda calculus may be consistently interpreted as domain descriptions: say, $id : \forall X.X \rightarrow X$ means that for each set or ”type” X, $id(X)$ is, in the model, a map from the set or ”type” X to itself. This is the only use of types in any model known up to now: a type input determines the type of the output, not the output itself. Such restriction to polymorphic maps is known as parametricity.

In this paper, we have shown that also a different interpretation is possible: types may be consistently intepreted as ”information-tags”, which are part of the term, and may be used in a definition by cases of a map. Here is an example of a map looking to the type-tag of the input to compute the output. Using the maps $Q', \mathsf{P}_{1,2}, j$ and test of proposition 4.1,4.3, we may define a map

$$\text{Newton} : \mathtt{Real} \rightarrow \forall X.(X \rightarrow (\mathtt{Real} + \mathtt{String}))$$

”Newton” takes a real x, a type X, an object $f : X$, and returns the result of applying, if possible (if $f : \mathtt{Real} \rightarrow \mathtt{Real} \rightarrow \ldots \rightarrow \mathtt{Real}$) the result of Newton algorithm to $x : \mathtt{Real}$ and to f. In the case f has not a type with the right shape, ”Newton” returns some string complaining it. We may write down the map ”Newton” using (fixed point and) the test map to test the shape of the type X, then Q', ¶ to ”disassembly” X, in order to check if X has the shape $\mathtt{Real} \rightarrow \mathtt{Real} \rightarrow \ldots \rightarrow \mathtt{Real}$.

We have thus shown that there exist a mathematical interpretation making sense of an use of typing, which could not be described in a model with only parametric polymorphic maps.

A Appendix: Proofs

We begin this appendix with the proof that $\Rrightarrow$ and $\mathtt{Q}$ (Definition 3.7) are well-defined and continuous. For such a proof we first need the following lemma.

Lemma A.1.

(i) $\vdash_{\mathtt{ho}}$ *and* $\vdash_{\mathtt{co}}$ *are reflexive and transitive.*
(ii) $(a^{\vdash_{\mathtt{co}}})^{\vdash_{\mathtt{co}}} = a^{\vdash_{\mathtt{co}}}$*;* $\quad (a^{\vdash_{\mathtt{ho}}})^{\vdash_{\mathtt{ho}}} = a^{\vdash_{\mathtt{ho}}}$
(iii) $x \,\mathtt{ho}\, y, x \vdash_{\mathtt{ho}} x', y \vdash_{\mathtt{ho}} y' \;\Rightarrow\; x' \,\mathtt{ho}\, y'$*.*
(iv) $x \,\mathtt{co}\, y, x \vdash_{\mathtt{co}} x', y \vdash_{\mathtt{co}} y' \;\Rightarrow\; x' \,\mathtt{co}\, y'$*.*
(v) If $a \subseteq$ Cons *is homogeneus(coherent) then* $a^{\vdash_{\mathtt{ho}}}$ *(*$a^{\vdash_{\mathtt{co}}}$*) is homogeneus(coherent).*

Proof. (i) Easy, by simultaneous induction on the definitions of $\vdash_{\mathtt{ho}}$ and $\vdash_{\mathtt{co}}$.
(ii) Immediate by (i).
(iii) We proceed by induction on the proof of $x \vdash_{\mathtt{ho}} x'$.

- Base cases.
  Trivial, since, by definition of $\vdash_{\mathtt{ho}}$ and $\mathtt{ho}$, it follows that $x \equiv x'$ and $y \equiv y'$.
- Inductive case: $x \equiv \langle c, z\rangle$, $x' \equiv \langle c', z'\rangle$ with $c' \vdash_{\mathtt{ho}} c$ and $z \vdash_{\mathtt{ho}} z'$.
  By definition of $\vdash_{\mathtt{ho}}$ and $\mathtt{ho}$ we obtain that $y \equiv \langle d, t\rangle$, $y' \equiv \langle d', t'\rangle$ with $d' \vdash_{\mathtt{ho}} d$ and $t \vdash_{\mathtt{ho}} t'$, moreover $c \,\mathtt{ho}\, d \Rightarrow z \,\mathtt{ho}\, t$. What we have to prove is $\langle c', z'\rangle \,\mathtt{ho}\, \langle d', t'\rangle$, that is , by definition of $\mathtt{ho}$, $c' \,\mathtt{ho}\, d' \Rightarrow z' \,\mathtt{ho}\, t'$. Let us assume $c' \,\mathtt{ho}\, d'$ in order to show $z' \,\mathtt{ho}\, t'$. Since $c' \vdash_{\mathtt{ho}} c$ and $d' \vdash_{\mathtt{ho}} d$, for any $u \in c$ and $v \in d$ there exist $u' \in c'$ and $v' \in d'$ such that $u' \vdash_{\mathtt{ho}} u$ and $v' \vdash_{\mathtt{ho}} v$. Moreover, $u' \,\mathtt{ho}\, v'$ because $c' \,\mathtt{ho}\, d'$. Then it is possible to apply the induction hypothesis on $u' \,\mathtt{ho}\, v'$, $u' \vdash_{\mathtt{ho}} u$ and $v' \vdash_{\mathtt{ho}} v$, obtaining $u \,\mathtt{ho}\, v$ for for any $u \in c$ and $v \in d$. This means that $c \,\mathtt{ho}\, d$. From $c \,\mathtt{ho}\, d$ we can now obtain $z \,\mathtt{ho}\, t$ by using our hypothesis $c \,\mathtt{ho}\, d \Rightarrow z \,\mathtt{ho}\, t$. By recalling that $z \vdash_{\mathtt{ho}} z'$ and $t \vdash_{\mathtt{ho}} t'$, we can apply the induction hypothesis on $z \,\mathtt{ho}\, t$, $z \vdash_{\mathtt{ho}} z'$ and $t \vdash_{\mathtt{ho}} t'$, obtaining $z' \,\mathtt{ho}\, t'$.

(iv) We proceed by induction on the proof of $x \,\mathtt{co}\, x'$.

- Base case.
  Trivial, since, by definition of $\vdash_{\mathtt{co}}$ and $\mathtt{co}$, $x' \equiv x \equiv y \equiv y'$.
- First inductive case: $x \equiv (c, z)$, $x' \equiv (c', z')$, with $c \,\mathtt{co}\, c'$ and $z \,\mathtt{co}\, z'$.
  We can proceed as done in the induction case of the proof of (ii). It is enough to exchange the role of $()$ and $\langle\rangle$, and of $\vdash_{\mathtt{ho}}$ and $\vdash_{\mathtt{co}}$.
- Second inductive case: $x \equiv \langle c, z\rangle$, $x \equiv \langle c', z'\rangle$, with $c' \vdash_{\mathtt{ho}} c$ and $z \vdash_{\mathtt{co}} z'$.
  By definition of $\vdash_{\mathtt{co}}$ and $\mathtt{co}$ it necessarily follows that $y \equiv \langle d, t\rangle$, $y' \equiv \langle d', t'\rangle$ with $c' \vdash_{\mathtt{ho}} c$, $d' \vdash_{\mathtt{ho}} d$, $z \vdash_{\mathtt{co}} z'$ and $t \vdash_{\mathtt{co}} t'$, moreover $c \,\mathtt{ho}\, d \Rightarrow z \,\mathtt{co}\, t$. What we have to prove is $\langle c', z'\rangle \,\mathtt{co}\, \langle d', t'\rangle$, that is , by definition of $\mathtt{co}$, $c' \,\mathtt{ho}\, d' \Rightarrow z' \,\mathtt{co}\, t'$. Let us assume $c' \,\mathtt{ho}\, d'$ in

order to derive $z' \,\mathtt{co}\, t'$. By (ii) it is possible to infer $c \,\mathtt{ho}\, d$. Since we know that $c \,\mathtt{co}\, d \Rightarrow z \,\mathtt{co}\, t$, we can infer also $z \,\mathtt{co}\, t$. By applying the induction hypothesis on $z \,\mathtt{co}\, t$, $z \vdash_{\mathtt{co}} z'$ and $t \vdash_{\mathtt{co}} t'$, we obtain $z' \,\mathtt{co}\, t'$.

(v) Easy by (ii) and (iii).

Proposition A.2.

(i) $\Rrightarrow$ *and* Q *are well-defined.*
(ii) $\Rrightarrow$ *and* Q *are continuous.*

Proof. (i)($\Rrightarrow$). We have to show that $X \Rrightarrow Y \subseteq \mathit{Cons}$ and that $X \Rrightarrow Y$ is homogeneous and closed w.r.t $\vdash_{\mathtt{ho}}$. Let $(a, y), (a', y') \in X \Rrightarrow Y$. Since $X \in \mathtt{Types}$, X is homogeneous. Then also a and a' are homogeneous, being subsets of an homogeneous set. Since a and a' are coherent as well and $y, y' \in \mathit{Cons}$, by definition of *Cons* it follows that $(a, y), (a', y') \in \mathit{Cons}$. Moreover, $(a, y) \,\mathtt{ho}\, (a', y')$ because $a \,\mathtt{ho}\, a'$ and $y \,\mathtt{ho}\, y'$ (a and a' are subsets of an homogeneous set, and y and y' are elements of an homogeneous set.) Since $\vdash_{\mathtt{ho}}$ restricted to $X \Rrightarrow Y$ is the identity relation, it follows immediately that $X \Rrightarrow Y$ is closed w.r.t. $\vdash_{\mathtt{ho}}$.
(i)(Q). We have to show that $\mathsf{Q}(F) \subseteq \mathit{Cons}$ and that $\mathsf{Q}(F)$ is homogeneous and closed for $\vdash_{\mathtt{ho}}$. Let $\langle a, y\rangle, \langle a', y'\rangle \in \mathsf{Q}(F)$. $\mathsf{Q}(F) \subseteq \mathit{Cons}$ by definition. In order to prove that $\langle a, y\rangle \,\mathtt{ho}\, \langle a', y'\rangle$, let assume $a \,\mathtt{ho}\, a'$. By Remark 3.3, $a \cup a'$ is homogeneous. From the monotonicity of F (F being continuous) we infer that $y, y' \in F((a \cup a')^{\vdash_{\mathtt{ho}}})$, and hence $y \,\mathtt{ho}\, y'$. this means that $a \,\mathtt{ho}\, a' \Rightarrow y \,\mathtt{ho}\, y'$, that is, by definition of $\mathtt{ho}$, $\langle a, y\rangle \,\mathtt{ho}\, \langle a', y'\rangle$. To show the closure of $\mathsf{Q}(F)$ with respect to $\vdash_{\mathtt{ho}}$, let us assume $\langle a, y\rangle \in \mathsf{Q}(F)$ and $\langle a, y\rangle \vdash_{\mathtt{ho}} \langle a', y'\rangle$, that is $a' \vdash_{\mathtt{ho}} a$ and $y \vdash_{\mathtt{ho}} y'$. By definition, $y \in F(a^{\vdash_{\mathtt{ho}}})$. Now, since $a^{\vdash_{\mathtt{ho}}} \subseteq a'^{\vdash_{\mathtt{ho}}}$ and F is monotone, we have also $y \in F(a'^{\vdash_{\mathtt{ho}}})$. But $F(a'^{\vdash_{\mathtt{ho}}})$ is in $\mathtt{Types}$, hence closed w.r.t $\vdash_{\mathtt{ho}}$, and then $y' \in F(a'^{\vdash_{\mathtt{ho}}})$. We obtain what we wished, that is $\langle a', y'\rangle \in \mathsf{Q}(F)$, by noticing that, by Lemma A.1(iii), $a \vdash_{\mathtt{ho}} a'$ implies that a' is homogeneous, being a homogeneous.
(ii)($\Rrightarrow$) $\Rrightarrow$ is trivially monotone. Let $X = \bigcup_{i\in I} X_i$ where $\{X_i\}_{i\in I}$ is directed. Then $X \Rrightarrow Y = \{(a, y) \mid a \in \mathcal{P}_{|\mathtt{co}}(\bigcup_{i\in I} X_i), a \text{ finite }, y \in Y\}$. If a is finite, the fact that the l.u.b. of two elements of $\{X_i\}_{i\in I}$ is their union (which is still in $\{X_i\}_{i\in I}$) implies that from $a \in \mathcal{P}_{|\mathtt{co}}(\bigcup_{i\in I} X_i)$ we can infer that there exists $k \in I$ such that $a \in \mathcal{P}_{|\mathtt{co}}(X_k)$. Hence $(\bigcup_{i\in I} X_i) \Rrightarrow Y = \{(a, y) \mid a \in \mathcal{P}_{|\mathtt{co}}(\bigcup_{i\in I} X_i), a \text{ finite }, y \in Y\} = \bigcup_{i\in I}\{(a, y) \mid a \in \mathcal{P}_{|\mathtt{co}}(X_i), a \text{ finite }, y \in Y\} = \bigcup_{i\in I}(X_i \Rrightarrow Y)$.
Let now $Y = \bigcup_{i\in I} Y_i$ where $\{Y_i\}_{i\in I}$ is directed. It is immediate to check that $X \Rrightarrow \bigcup_{i\in I} Y_i = \bigcup_{i\in I}(X \Rrightarrow Y_i)$.
(ii)(Q) Q is trivially monotone. Let $F = \bigsqcup_{i\in I} F_i$ where $\{F_i\}_{i\in I}$ is directed in $[\mathtt{Types} \to \mathtt{Types}]$. $\mathsf{Q}(\bigsqcup_{i\in I} F_i) = \{\langle a, y\rangle \mid a \in \mathcal{P}_{|\mathtt{ho}}(\mathit{Cons}), a \text{ finite }, y \in (\bigsqcup_{i\in I} F_i)(a^{\vdash_{\mathtt{ho}}})\} = \{\langle a, y\rangle \mid a \in \mathcal{P}_{|\mathtt{ho}}(\mathit{Cons}), a \text{ finite }, y \in \bigcup_{i\in I}(F_i(a^{\vdash_{\mathtt{ho}}}))\} = \bigcup_{i\in I}\{\langle a, y\rangle \mid a \in \mathcal{P}_{|\mathtt{ho}}(\mathit{Cons}), a \text{ finite }, y \in F_i(a^{\vdash_{\mathtt{ho}}})\} = \bigcup_{i\in I} \mathsf{Q}(F_i)$.

We provide now the proofs that $(\texttt{Types}, \subseteq, \bigcup)$, $(\texttt{Terms}, \subseteq, \bigcup)$ and $(\| X \|, \subseteq, \bigcup)$ are Scott domains.

Proof of Proposition 3.6 We only consider the case of Types, the case of Terms being similar.

Let $\bot_{\texttt{Types}} = \emptyset (\equiv \emptyset^{\vdash_{\texttt{ho}}})$. Let $\{Y_i\}_{i \in I}$ be a directed subset of Types. If two elements of Types have a common upper bound then they are homogeneous with each other. This, togheter with the fact that the union of sets closed w.r.t. $\vdash_{\texttt{ho}}$ is still closed w.r.t. $\vdash_{\texttt{ho}}$, implies that $\bigcup_{i \in I} Y_i \in \texttt{Types}$. It is immediate to see that $\bigcup_{i \in I} Y_i$ is the least upper bound of $\{Y_i\}_{i \in I}$. Hence $(\texttt{Types}, \subseteq, \bigcup)$ is complete.
By definition of Types, given $Y \in \texttt{Types}$, $a_0^{\vdash_{\texttt{ho}}} \subseteq Y$ for any a_0 which is a finite subset of Y. Therefore, given a directed $\{Y_i\}_{i \in I}$ and a finite homogeneous subset a_0 of $Y = \bigcup_{i \in I} Y_i$, there exists a finite set of indexes $J \subseteq I$ such that $a_0^{\vdash_{\texttt{ho}}} \subseteq \bigcup_{j \in J} Y_j$. Now, since $\bigcup_{j \in J} Y_j \in \{Y_i\}_{i \in I}$, the set of the compact elements of Types is $\{a_0^{\vdash_{\texttt{ho}}} \mid a_0 \in \mathcal{P}^{fin}_{|\texttt{ho}}(\mathit{Cons})\}$.
Now it is easy to check that Types is algebraic and bounded complete.

Proof of Proposition 3.9 $\bot_{\|X\|} = \emptyset (\equiv \emptyset^X)$. Let $\{a_i\}_{i \in I}$ be a directed subset of $\| X \|$. If two elements of $\| X \|$ have a common upper bound then they are coherent with each other. This, togheter with the fact that the union of sets closed w.r.t. the restriction of $\vdash_{\texttt{ho}}$ to X is still closed w.r.t. the restriction of $\vdash_{\texttt{ho}}$ to X, implies that $\bigcup_{i \in I} a_i \in \| X \|$. It is immediate to see that $\bigcup_{i \in I} a_i$ is the least upper bound of $\{a_i\}_{i \in I}$. Hence $(\| X \|, \subseteq, \bigcup)$ is complete.
By definition of $\| X \|$, given $a \in \| X \|$, $a_0^X \subseteq a$ for any a_0 which is a finite subset of a. Therefore, given a directed set $\{a_i\}_{i \in I}$ and a finite coherent subset a_0 of X, with $a_0 \subseteq a = \bigcup_{i \in I} a_i$, there exists a finite set of indexes $J \subseteq I$ such that $a_0^{\vdash_{\texttt{ho}}} \subseteq \bigcup_{j \in J} a_j$. Now, since $\bigcup_{j \in J} a_j \in \{a_i\}_{i \in I}$, the set of the compact elements of $\| X \|$ is $\{a_0^X \mid a_0 \in \mathcal{P}^{fin}_{|\texttt{co}}(X)\}$.
Now it is easy to check that $\| X \|$ is algebraic and bounded complete.

Proof of Proposition 3.11 Let us start proving Item (i) of Proposition 3.11.
In order to show that $\| X \Rrightarrow Y \| \simeq [\| X \| \to \| Y \|]$ we define two functions, $(-)^{\Downarrow}$ and $(-)^{\Uparrow}$. We then will show that they are well-defined and that form an isomorphism pair.

Definition A.3. *Let* $X, Y \in \texttt{Types}$.

(i) We define $(-)^{\Downarrow} : [\| X \| \to \| Y \|] \to \| X \Rrightarrow Y \|$ *as follows:*
Let $f \in [\| X \| \to \| Y \|]$

$$f^{\Downarrow} =_{Def} \{(a, y) \mid a \in \mathcal{P}^{fin}_{|\texttt{co}}(X),\ y \in f(a^X)\}.$$

(ii) We define $(-)^{\Uparrow} : \| X \Rrightarrow Y \| \to [\| X \| \to \| Y \|]$ *as follows:*
Let $A \in \| X \Rrightarrow Y \|$, $a \in \| X \|$

$$A^{\Uparrow}(a) =_{Def} \{y \mid (a_0, y) \in A,\ a_0 \subseteq a,\ a_0 \textit{finite}\}$$

Proposition A.4. *$(-)^{\Downarrow}$ and $(-)^{\Uparrow}$ are well-defined.*

Proof. Let us begin with $(-)^{\Downarrow}$.

- $f^{\Downarrow}$ is a coherent subset of $X \Rrightarrow Y$.
  Let $(a,y),(a',y') \in f^{\Downarrow}$. By definition of co, we have to prove that $a \,\mathsf{co}\, a' \Rightarrow y \,\mathsf{co}\, y'$. Let us assume $a \,\mathsf{co}\, a'$. It is immediate to see that $a \cup a'$ is a finite coherent subset of X and that $a^X, (a')^X \subseteq (a \cup a')^X$. By definition of $f^{\Downarrow}$, $y \in f(a^X)$ and $y' \in f((a')^X)$ and hence, by monotonicity of f (f being continuous), $y, y' \in f((a \cup a')^X) \in \| Y \|$. Thus, by definition of $\| - \|$ we obtain $y \,\mathsf{co}\, y'$.
- $f^{\Downarrow}$ is closed with respect to the restriction of $\vdash_{\mathsf{co}}$ to $X \Rrightarrow Y$.
  Let $(a,y) \in f^{\Downarrow}$, $(a',y') \in X \Rrightarrow Y$ and $(a,y) \vdash_{\mathsf{co}} (a',y')$. By definition of $X \Rrightarrow Y$ we have that $a' \in \mathcal{P}^{fin}_{|\mathsf{co}}(X)$, then, in order to show that $(a',y') \in f^{\Downarrow}$, we need to show that $y' \in f((a')^X)$. To prove this fact, let us notice that, from $(a,y) \vdash_{\mathsf{co}} (a',y')$ it follows that $a' \vdash_{\mathsf{co}} a$. This means that $a^X \subseteq (a')^X$. By the definition of $f^{\Downarrow}$, we have $y \in f(a^X)$. Then, by the monotonicity of f, $y \in f((a')^X)$. We can now infer $y' \in f((a')^X)$ from the fact that $f((a')^X) \in \| Y \|$. In fact $\| Y \|$ is closed with respect to the restriction of $\vdash_{\mathsf{co}}$ to Y by definition, and from $(a,y) \vdash_{\mathsf{co}} (a',y')$ it follows $y \vdash_{\mathsf{co}} y'$.

Now, let us proceed with $(-)^{\Uparrow}$.

- $A^{\Uparrow}$ maps $\| X \|$ into $\| Y \|$
  It is enough to show, for $a \in \| X \|$, $A^{\Uparrow}(a)$ to be a coherent subset of Y closed with respect to the restriction of $\vdash_{\mathsf{co}}$ to Y. $A^{\Uparrow}(a) \subseteq Y$ is immediate by definition. To show the coherence of $A^{\Uparrow}(a)$, let $y, y' \in A^{\Uparrow}(a)$. By definition, there exist $(a_0, y), (a_0', y') \in A$ such that $a_0, a_0' \subseteq a$. It then follows $a_0 \,\mathsf{co}\, a_0'$ because, by definition of $\|$ - $\|$, a is coherent. It is now possible to infer $y \,\mathsf{co}\, y'$ from the definition of co since $(a_0, y) \,\mathsf{co}\, (a_0', y')$. In order to show the closure of $A^{\Uparrow}(a)$ with respect to the restriction of $\vdash_{\mathsf{co}}$ to Y, let us assume $y \in A^{\Uparrow}(a)$, $y' \in Y$ and $y \vdash_{\mathsf{co}} y'$. By definition there exists $(a_0, y) \in A$. Since $a_0 \vdash_{\mathsf{co}} a_0$, from $y \vdash_{\mathsf{co}} y'$ we can infer $(a_0, y) \vdash_{\mathsf{co}} (a_0, y')$. This means that $(a_0, y') \in A$, since A is closed with respect to the restriction of $\vdash_{\mathsf{co}}$ to $X \Rrightarrow Y$, and $(a_0, y') \in X \Rrightarrow Y$. Then $y' \in A^{\Uparrow}(a)$ by definition of $A^{\Uparrow}(a)$.
- $A^{\Uparrow}$ is continuous.
  It is straightforward to check that $A^{\Uparrow}$ is monotone. Let $\{a_i\}_{i \in I}$ be a directed set in $\| X \|$ and $a = \bigcup_{i \in I} a_i$. $\bigcup_{i \in I} A^{\Uparrow}(a_i) \subseteq A^{\Uparrow}(a)$ by monotonicity. To prove the inverse relation, let $y \in A^{\Uparrow}(a)$. This means that there exists $(a_0, y) \in A$ with $a_0 \subseteq a$. Then there exists $k \in I$ such that $a_0 \subseteq (a_0)^X \subseteq a_k \subseteq a$, because $a = \bigcup_{i \in I} a_i$ and $(a_0)^X$ is a compact element of $\| X \|$. This means, by definition of $A^{\Uparrow}$, that $y \in A^{\Uparrow}(a_k)$ and hence $y \in \bigcup_{i \in I} A^{\Uparrow}(a_i)$.

Now we have to show that $(-)^{\Downarrow}$ and $(-)^{\Uparrow}$ form an isomorphism pair. In order to do that we need a couple of technical lemmas.

By definition of Scott domain it is straightforward to check the following property.

Proposition A.5. *Let D_1 and D_2 be two Scott domains such that the elements of D_2 are sets with the set theoretical union as l.u.b. operator. Then, given $f \in [D_1 \to D_2]$, $d \in D_1$ and $y \in f(d)$, there exists a compact element d_0 such that $d_0 \sqsubseteq d$ and $y \in f(d_0)$.*

Corollary A.6.

(i) *Let $f \in [\| X \| \to \| Y \|]$, $a \in \| X \|$ and $y \in f(a)$. Then there exists a_0 finite such that $a_0 \subseteq a$ and $y \in f(a_0^X)$.*

(ii) *Let $F \in [\mathtt{Types} \to \mathtt{Types}]$, $f \in [\prod_{T \in \mathtt{Types}} \| F(T) \|]$, $Y \in \mathtt{Types}$ and $y \in f(Y)$. Then there exists a_0 finite such that $a_0 \subseteq Y$ and $y \in f(a_0^{\vdash_{\mathtt{ho}}})$.*

(iii) *Let $F \in [\mathtt{Types} \to \mathtt{Types}]$, $Y \in \mathtt{Types}$ and $y \in F(Y)$. Then there exists a_0 finite such that $a_0 \subseteq Y$ and $y \in F(a_0^{\vdash_{\mathtt{ho}}})$.*

Lemma A.7. *(i) Let $A \in \| X \Rightarrow Y \|$ and $a \in \mathcal{P}^{fin}_{|\mathtt{co}}(X)$.*
Then $((a_0, y) \in A, a_0 \subseteq a^X) \;\;\Rightarrow\;\; (a, y) \in A$.

(ii) Let $A \in \| \mathsf{Q}(F) \|$ and $a \in \mathcal{P}^{fin}_{|\mathtt{ho}}$(Cons).
Then ($\langle a_0, y\rangle \in A, a_0 \subseteq a^{\vdash_{\mathtt{ho}}}) \;\;\Rightarrow\;\; \langle a, y\rangle \in A$.

Proof. (i) Since a^X is closed with respect to the restriction of $\vdash_{\mathtt{co}}$ to X, we have that $a_0 \subseteq a^X$ implies $a \vdash_{\mathtt{co}} a_0$. Then, by definition of $\vdash_{\mathtt{co}}$, $(a_0, y) \vdash_{\mathtt{co}} (a, y)$. This means that $(a, y) \in A$, since A is closed with respect to the restriction of $\vdash_{\mathtt{co}}$ to $X \Rightarrow Y$.
and hence, by definition of Scott domain, there exists a_0 finite, (ii) Since $a^{\vdash_{\mathtt{ho}}}$ is closed with respect $\vdash_{\mathtt{ho}}$, we have that $a_0 \subseteq a^{\vdash_{\mathtt{ho}}}$ implies $a \vdash_{\mathtt{ho}} a_0$. Then, by definition of $\vdash_{\mathtt{co}}$, $\langle a_0, y\rangle \vdash_{\mathtt{co}} \langle a, y\rangle$. This means that $\langle a, y\rangle \in A$, since A is closed with respect to the restriction of $\vdash_{\mathtt{co}}$ to $\mathsf{Q}(F)$.

Proposition A.8.

(i) $((-)^{\Downarrow})^{\Uparrow} = Id_{[\|X\| \to \|Y\|]}$.

(ii) $((-)^{\Uparrow})^{\Downarrow} = Id_{\|X \Rightarrow Y\|}$.

(iii) *Let $f, g \in [\| X \| \to \| Y \|]$. $\quad f \sqsubseteq g \;\;\Leftrightarrow\;\; f^{\Downarrow} \sqsubseteq g^{\Downarrow}$.*

Proof. (i) Let $f \in [\| X \| \to \| Y \|]$ and $a \in \| X \|$.
$(f^{\Downarrow})^{\Uparrow}(a) =$
$\{y \mid (a_0, y) \in \{(b, x) \mid b \in \mathcal{P}^{fin}_{|\mathtt{co}}(X),\ x \in f(b^X)\},\ a_0 \subseteq a\} =$
$\{y \mid y \in f(a_0^X),\ a_0 \subseteq a, a_0$ finite$\} = f(a)$ By Corollary A.6(i)

(ii) Let $A \in \| X \Rrightarrow Y \|$.
$(A^{\Uparrow})^{\Downarrow} =$
$\{(a, y) \mid a \in \mathcal{P}^{fin}_{|\mathtt{co}}(X),\ y \in \{y' \mid (a'_0, y') \in A,\ a'_0 \subseteq a^X\}\} =$
$\{(a, y) \mid a \in \mathcal{P}^{fin}_{|\mathtt{co}}(X),\ a'_0 \subseteq a^X,\ (a'_0, y) \in A\} = A$ By Lemma A.7(i)
(iii)($\Rightarrow$) In order to show $f^{\Downarrow} \subseteq g^{\Downarrow}$, let (a, y) such that $a \in \mathcal{P}^{fin}_{|\mathtt{co}}(X)$ and $y \in f(a^X)$. Since $f \sqsubseteq g$, we have that $f(a^X) \subseteq g(a^X)$ and hence $y \in g(a^X)$, that is $(a, y) \in g^{\Downarrow}$.
($\Leftarrow$) Towards a contradiction, let us assume $f \not\sqsubseteq g$. This means that there exist $a \in \| X \|$ and $y \in f(a)$ such that $y \notin g(a)$. By Corollary A.6(i), there exists $a_0 \subseteq a$ finite such that $(a_0, y) \in f^{\Downarrow}$.By the assumption $f^{\Downarrow} \subseteq g^{\Downarrow}$, $(a_0, y) \in g^{\Downarrow}$, and hence $y \in g(a_0^X)$. By monotonicity, $y \in g(a)$, contradiction.

Corollary A.9 (Proposition 3.11(i)).

$$\| X \Rrightarrow Y \| \simeq [\| X \| \to \| Y \|].$$

We can now pass to the proof of Item (ii) of Proposition 3.11 that is of the fact that $\| \mathsf{Q}(F) \| \simeq [\prod_{T \in \mathtt{Types}} \| F(T) \|]$.
Let us start by defining two functions, $(-)^{\Downarrow}$ and $(-)^{\Uparrow}$. Then we shall see that they are well-defined and that indeed they form an isomorphism pair.

Definition A.10. *(i) We define* $(-)^{\Downarrow} : [\prod_{T \in \mathtt{Types}} \| F(T) \|] \to \| \mathsf{Q}(F) \|$ *as follows:*
Let $f \in [\prod_{T \in \mathtt{Types}} \| F(T) \|]$

$$f^{\Downarrow} =_{Def} \{\langle a, y \rangle \mid a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\text{Cons}),\ y \in f(a^{\vdash_{\mathtt{ho}}})\}.$$

(ii) We define $(-)^{\Uparrow} : \| \mathsf{Q}(F) \| \to [\prod_{T \in \mathtt{Types}} \| F(T) \|]$ *as follows:*
Let $A \in \| \mathsf{Q}(F) \|$, $Y \in \mathtt{Types}$

$$A^{\Uparrow}(Y) =_{Def} \{y \mid \langle a_0, y \rangle \in A,\ a_0 \textit{ finite },\ a_0 \subseteq Y\}.$$

Proposition A.11. $(-)^{\Downarrow}$ *and* $(-)^{\Uparrow}$ *are well defined.*

Proof. Let us begin with $(-)^{\Downarrow}$.

– $f^{\Downarrow}$ is coherent.
Let $\langle a, y \rangle, \langle a', y' \rangle \in f^{\Downarrow}$. We need to show that $a \,\mathtt{ho}\, a' \Rightarrow y \,\mathtt{co}\, y'$. Let us assume $a \,\mathtt{ho}\, a'$. Then $a \cup a' \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons})$. Moreover, $a^{\vdash_{\mathtt{ho}}}, (a')^{\vdash_{\mathtt{ho}}} \subseteq (a \cup a')^{\vdash_{\mathtt{ho}}}$. By monotonicity, from $y \in f(a^{\vdash_{\mathtt{ho}}})$ and $y \in f((a')^{\vdash_{\mathtt{ho}}})$ we can infer $y, y' \in f((a \cup a')^{\vdash_{\mathtt{ho}}})$. We obtain $y \,\mathtt{co}\, y'$ by definition of $\| - \|$, since $f((a \cup a')^{\vdash_{\mathtt{ho}}}) \in \| F((a \cup a')^{\vdash_{\mathtt{ho}}}) \|$.
– $f^{\Downarrow}$ is closed w.r.t. the restriction of $\vdash_{\mathtt{co}}$ to $\mathsf{Q}(F)$.
Let $\langle a, y \rangle \in f^{\Downarrow}$, $\langle a, y \rangle \vdash_{\mathtt{co}} \langle a', y' \rangle$ with $\langle a', y' \rangle \in \mathsf{Q}(F)$. By definition of $\vdash_{\mathtt{co}}$, $a' \vdash_{\mathtt{ho}} a$. Hence $a^{\vdash_{\mathtt{ho}}} \subseteq (a')^{\vdash_{\mathtt{ho}}}$. By monotonicity and the fact that $y \in f(a^{\vdash_{\mathtt{ho}}})$, we obtain $y \in f((a')^{\vdash_{\mathtt{ho}}})$, that is $\langle a', y' \rangle \in f^{\Downarrow}$.

Now, let us proceed with $(-)^{\Uparrow}$.

- $A^{\Uparrow}(Y)$ is a coherent subset of $F(Y)$.
  - $A^{\Uparrow}(Y) \subseteq F(Y)$.

    Let $y \in A^{\Uparrow}(Y)$. This means that for a certain a_0 finite, $a_0 \subseteq Y$ and $\langle a_0, y\rangle \in A \subseteq \mathsf{Q}(F)$. By definition of Q we have $a_0 \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons})$ and $y \in F((a_0)^{\vdash_{\mathtt{ho}}})$. $y \in F(Y)$ is now a consequence of the monotonicity of F, because $a_0 \subseteq Y$ implies $(a_0)^{\vdash_{\mathtt{ho}}} \subseteq F(Y^{\vdash_{\mathtt{ho}}}) = F(Y)$.
  - $A^{\Uparrow}(Y)$ is coherent.

    Let $y, y' \in A^{\Uparrow}(Y)$. Then there exist $a_0, a'_0 \subseteq Y$ finite such that $\langle a_0, y\rangle, \langle a'_0, y'\rangle \in A$. Since A is coherent, $\langle a_0, y\rangle \,\mathtt{co}\, \langle a'_0, y'\rangle$ and hence, by definition of $\mathtt{co}$, $a_0 \,\mathtt{ho}\, a'_0 \Rightarrow y \,\mathtt{co}\, y'$. Is in now possible to infer $y \,\mathtt{co}\, y'$ since, being a_0 and a'_0 subsets of an homogeneous set, $a_0 \,\mathtt{ho}\, a'_0$.
- $A^{\Uparrow}(Y)$ is closed with respect to the restriction of $\vdash_{\mathtt{co}}$ to $F(Y)$.

  Let $y \in A^{\Uparrow}(Y)$, $y' \in F(Y)$ and $y \vdash_{\mathtt{co}} y'$. By definition there exists $a_0 \subseteq Y$ finite such that $\langle a_0, y\rangle \in A$. By Corollary A.6(iii) $y' \in F(Y)$ implies that there exists $a'_0 \subseteq Y$ finite such that $y' \in F((a'_0)^{\vdash_{\mathtt{ho}}})$. By monotonicity we get $y' \in F((a_0 \cup a'_0)^{\vdash_{\mathtt{ho}}})$. This means that $\langle a_0 \cup a'_0, y'\rangle \in \mathsf{Q}(F)$. Now, by definition of $\vdash_{\mathtt{ho}}$, $a_0 \cup a'_0 \vdash_{\mathtt{ho}} a_0$. Then, since $y \vdash_{\mathtt{co}} y'$, we get $\langle a_0, y\rangle \vdash_{\mathtt{co}} \langle a_0 \cup a'_0, y'\rangle$ by definition of $\vdash_{\mathtt{co}}$. Since A is closed with respect to the restriction of $\vdash_{\mathtt{co}}$ to $\mathsf{Q}(F)$, we obtain that $\langle a_0 \cup a'_0, y'\rangle \in A$ and, by definition of $A^{\Uparrow}(Y)$, $y' \in A^{\Uparrow}(Y)$.
- $A^{\Uparrow}$ is continuous.

  It is straightforward to check that $A^{\Uparrow}$ is monotone.

  Let $\{Y_i\}_{i\in I}$ be a directed set in $\mathtt{Types}$ and $Y = \bigcup_{i\in I} Y_i$. $\bigcup_{i\in I} A^{\Uparrow}(Y_i) \subseteq A^{\Uparrow}(Y)$ by monotonicity. To prove the inverse relation, let $y \in A^{\Uparrow}(Y)$. This means that there exists $\langle a_0, y\rangle \in A$ with $a_0 \subseteq Y$. Then there exists $k \in I$ such that $a_0 \subseteq (a_0)^{\mathsf{Q}(F)} \subseteq Y_k \subseteq Y$, since $Y = \bigcup_{i\in I} Y_i$ and $(a_0)^{\mathsf{Q}(F)}$ is a compact element of $|\mathsf{Q}(F)|$. This means, by definition of $A^{\Uparrow}$, that $y \in A^{\Uparrow}(Y_k)$ and hence $y \in \bigcup_{i\in I} A^{\Uparrow}(Y_i)$.

Proposition A.12.

(i) $((-)^{\Downarrow})^{\Uparrow} = Id_{[\prod_{T\in\mathtt{Types}}|F(T)|]}$.

(ii) $((-)^{\Uparrow})^{\Downarrow} = Id_{|\mathsf{Q}(F)|}$.

(iii) Let $f, g \in [\prod_{T\in\mathtt{Types}} |F(T)|]$. $\quad f \sqsubseteq g \;\Leftrightarrow\; f^{\Downarrow} \sqsubseteq g^{\Downarrow}$.

Proof. (i) Let $f \in [\prod_{T\in\mathtt{Types}} |F(T)|]$ and $Y \in \mathtt{Types}$.

$(f^{\Downarrow})^{\Uparrow}(Y) =$

$\{y \mid \langle a_0, y\rangle \in \{\langle b, x\rangle \mid b \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons}),\ x \in f(b^{\vdash_{\mathtt{ho}}})\},\ a_0 \subseteq Y\} =$

$\{y \mid y \in f((a_0)^{\vdash_{\mathtt{ho}}}),\ a_0$ finite $, a_0 \subseteq Y\} = f(Y)$ By Corollary A.6(ii)

(ii) Let $A \in \| \mathsf{Q}(F) \|$.
$(A^{\Uparrow})^{\Downarrow} =$
$\{\langle a, y\rangle \mid a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons}),\ y \in \{y' \mid \langle a'_0, y'\rangle \in A,\ a'_0 \subseteq a^{\vdash_{\mathtt{ho}}}\}\} =$
$\{\langle a, y\rangle \mid a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons}),\ \langle a'_0, y\rangle \in A,\ a'_0 \subseteq a^{\vdash_{\mathtt{ho}}}\} = A$ By Lemma A.7(ii) =
(iii) ($\Rightarrow$) In order to show $f^{\Downarrow} \sqsubseteq g^{\Downarrow}$, let $\langle a, y\rangle$ be such that $a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons})$ and $y \in f(a^{\vdash_{\mathtt{ho}}})$. Since $f \sqsubseteq g$, we have that $f(a^{\vdash_{\mathtt{ho}}}) \subseteq g(a^{\vdash_{\mathtt{ho}}})$ and hence $y \in g(a^{\vdash_{\mathtt{ho}}})$, that is $\langle a, y\rangle \in g^{\Downarrow}$.
($\Leftarrow$) Toward a contradiction, let us assume $f \not\sqsubseteq g$. This means that there exist $Y \in \mathtt{Types}$ and $y \in f(Y)$ such that $y \notin g(Y)$. By Corollary A.6(ii), there exists $a_0 \subseteq Y$ finite such that $\langle a_0, y\rangle \in f^{\Downarrow}$. By the assumption $f^{\Downarrow} \sqsubseteq g^{\Downarrow}$, $\langle a_0, y\rangle \in g^{\Downarrow}$, and hence $y \in g(a_0^{\vdash_{\mathtt{ho}}})$. It follows that, by monotonicity, $y \in g(Y)$, contradiction.

Corollary A.13 (Proposition 3.11(ii)).

$$\| \mathsf{Q}(F) \| \simeq [\prod_{T \in \mathtt{Types}} \| F(T) \|].$$

We can now pass to the proofs of the properties of our model stated in the Conclusions section.

Proof of Proposition 4.1 (i) Let $X, Y \in \mathtt{Types}$. We define

$$\mathsf{Q}'(X)(Y) =_{Def} \{y \mid \exists d \subseteq Y\ s.t.\ \langle d, y\rangle \in X\}.$$

Let us first show that Q' is well defined.

- $\mathsf{Q}'(X)(Y)$ is homogeneous.
  Let $y, y' \in \mathsf{Q}'(X)(Y)$. By definition there exists $d, d' \subseteq Y$ such that $\langle d, y\rangle, \langle d', y'\rangle \in X$. By the homogeneity of Y we infer $d \,\mathtt{ho}\, d'$, whereas the homogeneity of X implies $\langle d, y\rangle \,\mathtt{ho}\, \langle d', y'\rangle$. Then, by definition of $\mathtt{ho}$, $d \,\mathtt{ho}\, d' \Rightarrow y \,\mathtt{ho}\, y'$, and hence we can derive $y \,\mathtt{ho}\, y'$.
- $\mathsf{Q}'(X)(Y)$ is closed with respect to $\vdash_{\mathtt{ho}}$.
  Let $y \in \mathsf{Q}'(X)(Y)$ with $y \vdash_{\mathtt{ho}} y'$. By definition, there exists $d \subseteq Y$ such that $\langle d, y\rangle \in X$. by definition of $\mathtt{ho}$ we have that $\langle d, y\rangle \vdash_{\mathtt{ho}} \langle d, y'\rangle$. Since X is closed with respect to $\vdash_{\mathtt{ho}}$, we have also that $\langle d, y'\rangle \in X$, and hence $y' \in \mathsf{Q}'(X)(Y)$ by definition of Q'.

It is easy to check that Q' is continuous.
We can now prove $(\mathsf{Q}, \mathsf{Q}')$ to be a retraction pair for $[\mathtt{Types} \to \mathtt{Types}] \lhd \mathtt{Types}$.
Let $F \in [\mathtt{Types} \to \mathtt{Types}]$. By definition of Q' and Q we have
$\mathsf{Q}'(\mathsf{Q}(F))(Y) =$
$\{y \mid \exists d \subseteq Y\ s.t.\ \langle d, y\rangle \in \{\langle a, y'\rangle \mid a \in \mathcal{P}^{fin}_{|\mathtt{ho}}(\mathit{Cons}), y' \in F(a^{\vdash_{\mathtt{ho}}})\}\} =$

$\{y \mid \exists d \subseteq Y\ s.t.\ d$ finite, $y \in F(d^{\vdash_{\texttt{ho}}})\} = F(Y)$ By Corollary A.6(iii)

(ii) Let $X \in \texttt{Types}$. We define

$$\mathsf{P}_{1,2} =_{Def} (\mathsf{P}_1(X), \mathsf{P}_2(X))$$

where

$$\mathsf{P}_1(X) =_{Def} \bigcup \{a \mid \exists x.\ (a,x) \in X\}^{\vdash_{\texttt{ho}}},$$

$$\mathsf{P}_2(X) =_{Def} \{x \mid \exists a.\ (a,x) \in X\}^{\vdash_{\texttt{ho}}}.$$

Let us first prove that $\mathsf{P}_1(X), \mathsf{P}_2(X) \in \texttt{Types}$. $\mathsf{P}_1(X), \mathsf{P}_2(X) \subseteq$ *Cons* since $(a,x) \in X$ implies $a \subseteq$ *Cons* and $x \in$ *Cons*. $\mathsf{P}_1(X)$ and $\mathsf{P}_2(X)$ are homogeneous because X is so. In fact $(a,x),(a',x') \in X$ implies $a \,\texttt{ho}\, a'$ and $x \,\texttt{ho}\, x'$. The closure of $\mathsf{P}_1(X)$ and $\mathsf{P}_2(X)$ with respect to $\vdash_{\texttt{ho}}$ is trivial by definition.

Let us now show that $\mathsf{P}_{1,2} \circ \Rrightarrow = Id_{\texttt{Types}\times\texttt{Types}}$. Indeed, using the remark that $\{d\} \in \mathcal{P}_{|\texttt{co}}(X)$ for any $d \in X$, and the fact that X and Y are closed with respect to $\vdash_{\texttt{ho}}$, $\mathsf{P}_{1,2}(X \Rrightarrow Y) \equiv \mathsf{P}_{1,2}(\{(a,y) \mid a \in \mathcal{P}^{fin}_{|\texttt{co}}(X),\ y \in Y\}) \equiv (X^{\vdash_{\texttt{ho}}}, Y^{\vdash_{\texttt{ho}}} \equiv (X, Y)$.

To complete the proof let us prove $\mathsf{P}_{1,2}$ to be continuous. $\mathsf{P}_{1,2}$ is trivially monotone. Let $\{X_i\}_{i\in I}$ be directed. We have that $(x,y) \in \mathsf{P}_{1,2}(\bigcup_{i\in I} X_i)$ if and only if there exist x', y' such that $x' \vdash_{\texttt{ho}} x$, $y' \vdash_{\texttt{ho}} y$ and $(a,x'),(b,y') \in X_i$ with $x' \in a$ for some a, b and i. This means that $(x,y) \in \mathsf{P}_{1,2}(\bigcup_{i\in I} X_i)$ if and only if $(x,y) \in \mathsf{P}_{1,2}(X_i)$ for some i, that is $\mathsf{P}_{1,2}(\bigcup_{i\in I} X_i) = \bigcup_{i\in I} \mathsf{P}_{1,2}(X_i)$.

We give now the proof of the fact that our model does not satisfy the ”axiom C”.

Proof of Proposition 4.2 Axiom C is a (very) particular case of parametricity. It says: all polymorphic maps in types $\forall\alpha A$, with α not free in A, are constant in α. It may be expressed by saying: if $t, u : \forall\alpha A$, with α not free in A, then $t(B) =_C u(B)$, for any type B.

Let now $A = \forall\alpha,\beta,\gamma.((\forall\delta.\gamma) \to \gamma)$ and $t, u : A$ defined by $t = \lambda\alpha,\beta,\gamma.\lambda y : (\forall\delta.\gamma).y(\alpha)$ and $u = \lambda\alpha,\beta,\gamma.\lambda y : (\forall\delta.\gamma).y(\beta)$

t, u are equal if we assume axiom C. Indeed, if $y : \forall\delta.\gamma$, then δ is not free in γ.Thus, $y(\alpha) =_C y(\beta)$. Now abstract over $y : \forall\delta.\gamma$, then abstract over α,β,γ. Eventually, we get $t =_C u : A$.

Still, t, u are different in our model. This is a consequence of $\beta\eta$-completeness, and of the fact that t, u are not $\beta\eta$- convertible. Alternatively, we may check $t \neq u$ directly, by finding some $\alpha,\beta,\gamma \in \texttt{Types}$, and some $y : (\forall\delta.\gamma)$ in the model, such that $t(\alpha,\beta,\gamma,y) \neq u(\alpha,\beta,\gamma,y)$, that is, $y(\alpha) \neq y(\beta)$. Take, for instance, $\alpha = \emptyset, \beta = \{0\}, \gamma = \{0\}, y = \{\langle\{0\}, 0\rangle\}$, then check that $y(\alpha) = \emptyset, y(\beta) = \{0\}$.

Proof of Proposition 4.3 (i) Let $X, Y \in \texttt{Types}$ and $a \in |X|$. We define

$$\mathsf{j}(X,Y)(a) =_{Def} a^Y.$$

By definition of a^Y it is easy to check that a^Y is an element of Y and that $\mathsf{j}(X,Y)$ is continuous.

(ii) Let $l_i \in L_i$ be a choice of elements, one for each L_i. By assumption, the sets L_i are pairwise disjoint. By definition of homogeneity over L_i, it follows that the elements l_i are pairwise non-homogeneous.

Let now $X \in \texttt{Types}$. We define

$$\mathsf{test}(X) =_{Def} \begin{cases} \{0\} & \text{if } (a,x) \in X \text{ for some } a \text{ and } x; \\ \{1\} & \text{if } \langle a,x\rangle \in X \text{ for some } a \text{ and } x; \\ \{i+2\} & \text{if } l_i \in X \text{ for some } l_i \in \mathcal{L}; \\ \{\} & \text{otherwise} \end{cases}$$

test is well-defined. Indeed, by the homogeneity of X, the fact that $l_i, (a,x), \langle a,x\rangle$ are by definition non-homogeneous, the three first conditions are pairwise incompatible. The index i in the third condition is uniquely given, because the element l_i are pairwise incompatible. test is continuous because L_0 is a flat domain and test is monotone.

References

1. AMADIO R., BRUCE K.B., LONGO G. (1986) The Finitary Projection Model for Second-order Lambda-calculus, In *Logic in Computer Science*, IEEE Computer Society Press, 122–130.
2. BARENDREGT H., The λ-calculus, its syntax and semantics, Studies in Logic vol.103, North-Holland, revised edition 1984.
3. BERARDI S., Retractions on dI-Domains as a Model for Type:Type, Information and Computation 94, p.204-231, 1991.
4. BERARDI S., Ch. Berline, Building continuous webbed models for System F. To appear on: proceedings of MFPS 14.
5. BERARDI S., BERLINE C., β-η complete models for system F. Tech.report Dip. di Informatica, Universitá di Torino,1998.
6. BERLINE C., Rétractions et Interpretation Interne du Polymorphisme : le Problème de la Rétraction Universelle, Informatique théorique et Applications/Theoretical Informatics and Applications vol.26, n$^\circ 1$, p.59-91, 1992.
7. BRUCE K.B., MEYER A.R., MITCHELL J.C. (1990) The Semantics of Second-Order Lambda Calculus, *Information and Computation* 85, 76-134.
8. COQUAND T., A. GUNTER C.A., AND WINSKEL G. (1989) Domain Theoretic models of Polymorphism, *Information and Computation*, vol. 81 (1989), 123–167.
9. GIRARD J.Y. (1972) *Interprétation Fonctionelle et Élimination des Coupures de l'Arithmétique d'Ordre Superiéur*, Thése d'Etat, University of Paris VII.
10. GIRARD J.Y. (1986) The system F of variable types, fifteen years later, *Theoretical Computer Science*, vol.45, 159–192.
11. LONGO G., MILSTED K., SOLOVIEV S., The Genericity Theorem and effective Parametricity in Polymorphic lambda-calculus. *Theoretical Computer Science*, 121:323–349, 1993.
12. MCCRACKEN N. A Finitary Retract Model for the Polymorphic Lambda-calculus, unpublished, 1984.
13. MITCHELL J.C. *Foundations for Programming Languages*, The MIT Press, 1996.

14. PITTS A., Polymorphism is set-theoretic, constructively, In *Category Theory and Computer Science*, LNCS 283, Spinger-Verlag, 1987.
15. REYNOLDS J.C. Towards a theory of type structure, In *Paris Programming symposium*, LNCS 19, Springer-Verlag, 157–168, 1974.
16. REYNOLDS J.C. Polymorphism is not set-theoretic, In *Semantics of Data Types*, LNCS 173, Spinger-Verlag, 145–156, 1984.