# Tailoring Filter Models

F. Alessi[1], F. Barbanera[2*], and M. Dezani-Ciancaglini[3**]

[1] Dipartimento di Matematica e Informatica, Via delle Scienze, 206 33100 Udine (Italy)
alessi@dimi.uniud.it
[2] Dipartimento di Matematica e Informatica, Viale A.Doria, 6 95125 Catania (Italy)
barba@dmi.unict.it
[3] Dipartimento di Informatica, Corso Svizzera, 125 10149 Torino (Italy)
dezani@di.unito.it

**Abstract.** Conditions on type preorders are provided in order to characterize the induced filter models for the $\lambda$-calculus and some of its restrictions. Besides, two examples are given of filter models in which not all the continuous functions are representable.

## 1. Introduction

The semantics of the $\lambda$-calculus can be looked at from several points of view. A possible one considers a model as an abstract way of handling and dealing with the syntax. This is the point of view of those investigations looking for extensions of the $\lambda$-calculus such that the intended semantical domain turns out to be fully abstract w.r.t. the calculus.

From another point of view, instead, the semantics is seen mainly as a tool to confirm one's "syntactic intuitions" and to prove properties of the calculus. According to this latter viewpoint, "semantically oriented" extensions of a calculus are not always commendable. The focus is on the calculus: the model has to fit as tight as possible the calculus, not vice versa. This is indeed the point of view of the present paper, and, in general, the one of an investigation we are carrying on, started in a companion paper [3]. In such a research we try to devise a general setting and uniform tools to "tailor" models closely fitting as many as possible aspects of the computational paradigm embodied by the $\lambda$-calculus.

One of the most natural framework for such an investigation is the typing discipline with *Intersection Types*. Intersection type assignment systems allow to characterize many of the most important *denotational* (as well as *operational*) properties of $\lambda$-terms. In particular it is possible to describe, in a *natural* and *finitary* way, many semantic domains for the $\lambda$-calculus. Such finitary descriptions allow not only to analyze pre-existing models, but also to modify them, sometimes "tailoring" them according to one's needs (see [6, 10, 14, 18, 17, 22, 5, 12] and the references there.)

Finitary characterizations of models for the $\lambda$-calculus, the so called *filter models*, can be obtained by simply introducing specific constants, typing rules and type preorders in a basic intersection type assignment system. An element of a particular domain, representing the denotational meaning of a term $M$, comes then out to correspond

[*] Partially supported by MURST project NAPOLI.

[**] Partially supported by EU within the FET - Global Computing initiative, project DART ST-2001-33477, and by MURST Cofin'02 project McTati. The funding bodies are not responsible for any use that might be made of the results presented here.

to the set of types that can be inferred for $M$.
In [3] we have characterized those intersection type assignment systems aiming, in perspective, at providing finitary descriptions of filter models validating in a precise way the notions of $\beta$ and $\eta$ reduction and expansion for the whole $\lambda$-calculus, as well as some of their restrictions, like $\beta_v$ [21], $\beta$-**I** [11] and $\beta$-**K**N [17].

The present paper keeps on the same direction by proving a number of characterization results for filter $\lambda$-structures induced by type preorders.
Since any type preorder can induce a particular filter $\lambda$-structure, it is possible to "tailor" particular models by providing suitable conditions on the inducing type preorders. Our first "tailoring" result characterizes those type preorders inducing $\lambda$-structures in which relevant sets of functions can be represented. A second result characterizes $\lambda$-structures which are models of the whole $\lambda$-calculus. In a third result we characterize those filter $\lambda$-structures which are also models of the aforementioned restricted $\lambda$-calculi: the call-by-value $\lambda$-calculus, the $\lambda$**I**-calculus, the $\lambda$**K**N-calculus. The result is also extended to the extensional models.

A further "tailoring" result of the present paper concerns the possibility of "trimming" something that is usually overabundant in filter models: the set of the representable functions. Such a task is not a trivial one in the intersection filters setting. In fact in any filter model introduced in the literature, but the one in [8], any continuous function is representable. Our contribution to this task is the construction of type preorders inducing filter models of the whole $\lambda$-calculus in which not all continuous functions are representable. The proofs of this property will profit from the characterization results of the paper.

We shall assume the reader to be acquainted with the main concepts concerning the $\lambda$-calculus and its models. The paper will be structured as follows: in Section 2 we recall the notions of intersection type language, type preorder and type assignment system, while the definitions of filter $\lambda$-structure and filter model will be recalled in Section 3. The four characterization results will form the subject of Section 4. In Section 5 we shall define two particular preorders in whose induced filter models only a proper subset of the continuous functions is representable.

## 2. Intersection types languages and type assignments

*Intersection types*, the building blocks for the filter models, are syntactical objects built by closing a given set $\mathbb{C}$ of *type atoms* (constants) under the *function type* constructor $\rightarrow$ and the *intersection type* constructor $\cap$.

**Definition 1 (Intersection type language).** *The* intersection type language *over* $\mathbb{C}$, *denoted by* $\mathbb{T} = \mathbb{T}(\mathbb{C})$ *is defined by the following abstract syntax:*

$$\mathbb{T} = \mathbb{C} \mid \mathbb{T}\rightarrow\mathbb{T} \mid \mathbb{T} \cap \mathbb{T}.$$

Much of the expressive power of intersection type languages comes from the fact that they are endowed with a *preorder relation*, $\leq$, which induces, on the set of types, the structure of a meet semi-lattice with respect to $\cap$.

**Definition 2 (Intersection type preorder).** *An* intersection type preorder *is a pair* $(\mathbb{C}, \leq)$ *where* $\mathbb{C}$ *is a set of type constants and* $\leq$ *is a binary relation over* $\mathbb{T} = \mathbb{T}(\mathbb{C})$

*satisfying the following set of axioms and rules:*

$$\begin{array}{rlrl}
\textit{(refl)} & A \leq A & \textit{(idem)} & A \leq A \cap A \\
\textit{(incl}_L\textit{)} & A \cap B \leq A & \textit{(incl}_R\textit{)} & A \cap B \leq B \\
\textit{(mon)} & \dfrac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'} & \textit{(trans)} & \dfrac{A \leq B \quad B \leq C}{A \leq C} \\
(\Omega)\ \textit{if}\ \Omega{\in}\mathbb{C} & A \leq \Omega & (\nu)\ \textit{if}\ \nu{\in}\mathbb{C} & A{\to}B \leq \nu
\end{array}$$

NOTATION. - $\Sigma$ will be short for $(\mathbb{C}, \leq)$.
- $A \sim B$ will be short for $A \leq B \leq A$.
- Since $\cap$ is commutative and associative (modulo $\sim$), we shall write $\bigcap_{i \leq n} A_i$ for $A_1 \cap \ldots \cap A_n$. Similarly we shall write $\cap_{i \in I} A_i$, where $I$ denotes always a finite set. Moreover we make the convention that $\cap_{i \in \emptyset} A_i$ is $\Omega$ when $\Omega{\in}\mathbb{C}$.
- We shall denote by $\leq_{\nabla}$ the type preorder generated by a recursive set $\nabla$ of axioms and rules of the shape $A \leq B$ (where $\nabla$ it is said to generate $\leq$ if $A \leq B$ holds if and only if it can be derived from the axioms and rules of $\nabla$ together with those in Definition 2). The constants in $\nabla$ will be denoted by $\mathbb{C}^{\nabla}$.
- When we consider an intersection type preorder of the form $(\mathbb{C}^{\nabla}, \leq_{\nabla})$, we shall write $\mathbb{T}^{\nabla}$ and $\Sigma^{\nabla}$ for $\mathbb{T}(\mathbb{C}^{\nabla})$ and $(\mathbb{C}^{\nabla}, \leq_{\nabla})$, respectively.
- $A {\sim_{\nabla}} B$ will be short for $A \leq_{\nabla} B \leq_{\nabla} A$.
- We write "the type preorder $\Sigma$ validates $\nabla$" to mean that all axioms and rules of $\nabla$ are admissible.[1]

Figure 1 lists a few special purpose axioms and rules which have been considered in the literature. Their meaning can be grasped if we consider types to denote subsets of a domain of discourse and we look at → as the function space constructor in the light of Curry-Scott semantics, see [23].

| | | | |
|---|---|---|---|
| $(\Omega\text{-}\eta)$ | $\Omega \leq \Omega{\to}\Omega$ | $({\to}\text{-}{\cap})$ | $(A{\to}B) \cap (A{\to}C) \leq A{\to}B \cap C$ |
| $(\Omega\textit{-lazy})$ | $A{\to}B \leq \Omega{\to}\Omega$ | $(\eta)$ | $\dfrac{A' \leq A \quad B \leq B'}{A{\to}B \leq A'{\to}B'}$ |

**Fig. 1.** Possible Axioms and Rules concerning $\leq$.

We can introduce now four significant intersection type preorders which have been extensively considered in the literature. The order is logical, rather than historical, and the references define the corresponding filter models: [9], [14], [1], [6]. A richer list of type preorders can be found in [3]. These preorders are of the form $\Sigma^{\nabla} = (\mathbb{C}^{\nabla}, \leq_{\nabla})$, with various different names $\nabla$, picked for mnemonic reasons. In Figure 2 we list their sets of constants $\mathbb{C}^{\nabla}$ and their sets $\nabla$ of extra axioms and rules taken from Figure 1. Here $\mathbb{C}_{\infty}$ is an infinite set of fresh atoms (i.e. different from $\Omega, \nu$).

[1] Recall that a rule is *admissible* in a system if, for each instance of the rule, if its premises are derivable in the system then so is its conclusion.

| | |
|---|---|
| $\mathbb{C}^{\mathcal{CDV}} = \mathbb{C}_\infty$ | $\mathcal{CDV} = \{(\to\text{-}\cap), (\eta)\}$ |
| $\mathbb{C}^{\mathcal{EHR}} = \{\nu\}$ | $\mathcal{EHR} = \mathcal{CDV} \cup \{(\nu)\}$ |
| $\mathbb{C}^{\mathcal{AO}} = \{\Omega\}$ | $\mathcal{AO} = \mathcal{CDV} \cup \{(\Omega), (\Omega\text{-}lazy)\}$ |
| $\mathbb{C}^{\mathcal{BCD}} = \{\Omega\} \cup \mathbb{C}_\infty$ | $\mathcal{BCD} = \mathcal{CDV} \cup \{(\Omega), (\Omega\text{-}\eta)\}$ |

**Fig. 2.** Particular Atoms, Axioms and Rules.

### 2.1. Particular classes of type preorders

In this subsection we introduce important classes of type preorders. The first two are the classes of *natural* type preorders and of *strict natural* type preorders. These are *disjoint* classes, whose relevance lies in their allowing various characterizations in terms of approximable mappings and $\lambda$-structures.

**Definition 3 ((Strict) Natural type preorders).**
*Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder.*

(i) *$\Sigma$ is* strict natural *if $\Omega \notin \mathbb{C}$ and it validates $\mathcal{CDV}$ as defined in Figure 2.*

(ii) *$\Sigma$ is* natural *if $\Omega \in \mathbb{C}$ and it validates $\mathcal{AO}$ as defined in Figure 2.*

Naturality for type preorders has a strong semantic flavour. If we look at intersection as representing join and at arrow types as representing functions, then rule ($\to$-$\cap$) reflects the join property of step functions with the same antecedent ($(d \Rightarrow e) \sqcup (d \Rightarrow e') \sqsupseteq d \Rightarrow (e \sqcup e')$), rule ($\eta$) reflects the order relation between step functions ($d' \sqsubseteq d$ and $e \sqsubseteq e'$ imply $d \Rightarrow e \sqsubseteq d' \Rightarrow e'$), and rule ($\Omega$-*lazy*) reflects the fact that $\bot \Rightarrow \bot$ is the bottom function.

Among the type preorders of Figure 2, $\mathcal{CDV}$, $\mathcal{EHR}$ are strict natural, and $\mathcal{AO}$, $\mathcal{BCD}$ are natural.

Notice that by the implicit assumption that axiom $(\nu) \in \Sigma$ whenever $\nu \in \mathbb{C}^{\Sigma}$ (Definition 2) a strict natural type theory containing the constant $\nu$ validates $\mathcal{EHR}$.

We introduce two other interesting classes of preorders playing a crucial role in the characterization results of Section 4.

**Definition 4 (Beta and eta preorders).**
*Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder and $\mathbb{T} = \mathbb{T}(\mathbb{C})$.*

(i) *$\Sigma$ is* beta *iff for all $I$, $A_i, B_i, C, D \in \mathbb{T}$:*
$\bigcap_{i \in I}(A_i \to B_i) \leq C \to D \implies \bigcap_{i \in J} B_i \leq D$ *where* $J = \{i \in I \mid C \leq A_i\}$.

(ii) *$\Sigma$ is* eta *iff for all $\psi \in \mathbb{C}$ at least one of the following conditions hold:*

*(1) $\nu \leq \psi$;*

*(2) there exist $I, A_i, B_i \in \mathbb{T}$ such that $\bigcap_{i \in I}(A_i \to B_i) \leq \psi$ and $B_i \sim \Omega$ for all $i \in I$;*

*(3) there exist non empty families of types $\{A_i, B_i\}_{i \in I}$, $\{D_{i,j}, E_{i,j}\}_{j \in J_i}$ in $\mathbb{T}$ such that*

$$\bigcap_{i \in I}(A_i \to B_i) \leq \psi \leq \bigcap_{i \in I}(\bigcap_{j \in J_i}(D_{i,j} \to E_{i,j})) \;\&$$
$$\forall i \in I.\; A_i \leq \bigcap_{j \in J_i} D_{i,j} \;\&\; \bigcap_{j \in J_i} E_{i,j} \leq B_i.$$

The condition for a natural type theory of being beta reflects the criterion used to establish if a sups of step functions is greater than a step function (see [15]).

When $\Sigma = \Sigma^{\nabla}$ it is usually possible to prove the conditions defined above by induction on the derivation of judgments. For the type preorders of Figure 2 we get that they are all beta and $\mathcal{EHR}, \mathcal{AO}$ are eta.

### 2.2. Intersection type assignments

We introduce now the notion of *intersection type assignment system*. First we need a few preliminary definitions. Let *Var* denote the set of term variables.

**Definition 5 (Type assignment system).**
(i) *A* basis *over* $\mathbb{C}$ *is a set of statements of the shape* $x{:}B$*, where the* subjects $x$ *are in Var, the* predicates $B$ *are in* $\mathbb{T}(\mathbb{C})$*, and all subjects are distinct variables.*
(ii) *An* intersection-type assignment system *relative to* $\Sigma = (\mathbb{C}, \leq)$*, denoted by* $\lambda\cap^{\Sigma}$*, is a formal system for deriving judgments of the form* $\Gamma \vdash^{\Sigma} M : A$*, where the* subject $M$ *is an untyped* $\lambda$*-term, the* predicate $A$ *is in* $\mathbb{T}(\mathbb{C})$*, and* $\Gamma$ *is a basis over* $\mathbb{C}$*.*

NOTATION. We shall write:
- $x{\in}\Gamma$ as short for $(x : A){\in}\Gamma$ for some $A$;
- $\Gamma, x{:}A$ as short for $\Gamma \cup \{x{:}A\}$, proviso $x{\notin}\Gamma$.

We use $\uplus$ to denote the union between bases defined by:

$$\begin{aligned}\Gamma_1 \uplus \Gamma_2 = {} & \{(x{:}\tau) \mid (x{:}\tau){\in}\Gamma_1 \& x{\notin}\Gamma_2\} \cup \\ & \{(x{:}\tau) \mid (x{:}\tau){\in}\Gamma_2 \& x{\notin}\Gamma_1\} \cup \\ & \{(x{:}\tau_1 \cap \tau_2) \mid (x{:}\tau_1){\in}\Gamma_1 \& (x{:}\tau_2){\in}\Gamma_2\}\end{aligned}$$

A term $M$ is said to be *typable* in $\lambda\cap^{\Sigma}$, for a given basis $\Gamma$, if there is a type $A{\in}\mathbb{T}(\mathbb{C})$ such that the judgment $\Gamma \vdash^{\Sigma} M : A$ is derivable.

Various type assignment systems can be defined, each of them parametrized w.r.t a particular $\Sigma$=$(\mathbb{C}, \leq)$. The simplest system is given in the following definition.

**Definition 6 (Basic type assignment system).**
*Given a type preorder* $\Sigma$*, the axioms and rules of the* basic type assignment system, *denoted by* $\lambda\cap^{\Sigma}_{\mathcal{B}}$*, for deriving judgments* $\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A$*, are the following:*

$$(Ax) \quad \Gamma \vdash^{\Sigma}_{\mathcal{B}} x{:}A \qquad \text{if } (x{:}A){\in}\Gamma$$

$$(\to\mathrm{I})\ \frac{\Gamma, x{:}A \vdash^{\Sigma}_{\mathcal{B}} M : B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} \lambda x.M : A{\to}B} \qquad (\to\mathrm{E})\ \frac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \to B \quad \Gamma \vdash^{\Sigma}_{\mathcal{B}} N : A}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} MN : B}$$

$$(\cap\mathrm{I})\ \frac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \quad \Gamma \vdash^{\Sigma}_{\mathcal{B}} M : B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \cap B} \quad (\leq)\ \frac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \quad A \leq B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : B}$$

If $\Omega{\in}\mathbb{C}$, a natural choice is to set $\Omega$ as the universal type of all $\lambda$-terms. This amounts to modify the basic type assignment system by adding a suitable axiom for $\Omega$.

**Definition 7 ($\Omega$-type assignment system).**
*Given a type preorder* $\Sigma = (\mathbb{C}, \leq)$ *with* $\Omega{\in}\mathbb{C}$*, the axioms and rules of the* $\Omega$-type assignment system *(denoted* $\lambda\cap^{\Sigma}_{\Omega}$*), for deriving judgments of the form* $\Gamma \vdash^{\Sigma}_{\Omega} M : A$*, are those of the basic one, plus the further axiom*

$$(Ax\text{-}\Omega)\ \Gamma \vdash^{\Sigma}_{\Omega} M : \Omega.$$

Analogously to the case of $\Omega$, when $\nu\in\mathbb{C}$, it is natural to consider $\nu$ as the universal type for abstractions, hence modifying the basic system by the addition of a special axiom for $\nu$.

**Definition 8 ($\nu$-type assignment system).**
*Given a type preorder $\Sigma = (\mathbb{C}, \leq)$ with $\nu\in\mathbb{C}$, the axioms and rules of the $\nu$-*type assignment system *(denoted $\lambda\cap_\nu^\Sigma$), for deriving judgements of the form $\Gamma \vdash_\nu^\Sigma M : A$, are those of the basic one, plus the further axiom*

$$(\mathit{Ax}\text{-}\nu)\ \ \Gamma \vdash_\nu^\Sigma \lambda x.M : \nu.$$

For simplicity we assume the symbols $\Omega$ and $\nu$ to be reserved for the universal type constants respectively used in the systems $\lambda\cap_\Omega^\Sigma$ and $\lambda\cap_\nu^\Sigma$, i.e. we forbid $\Omega\in\mathbb{C}$ or $\nu\in\mathbb{C}$ when we deal with $\lambda\cap_{\mathcal{B}}^\Sigma$.

NOTATION. - $\lambda\cap^\Sigma$ will range over $\lambda\cap_{\mathcal{B}}^\Sigma$, $\lambda\cap_\Omega^\Sigma$ and $\lambda\cap_\nu^\Sigma$. More precisely we assume that $\lambda\cap^\Sigma$ stands for $\lambda\cap_\Omega^\Sigma$ whenever $\Omega\in\mathbb{C}$, for $\lambda\cap_\nu^\Sigma$ whenever $\nu\in\mathbb{C}$, and for $\lambda\cap_{\mathcal{B}}^\Sigma$ otherwise. Similarly for $\vdash^\Sigma$.
- When $\Sigma = \Sigma^\nabla$ we shall denote $\lambda\cap^\Sigma$ and $\vdash^\Sigma$ by $\lambda\cap^\nabla$ and $\vdash^\nabla$, respectively.

It is easy to prove that the following rules are admissible in $\lambda\cap^\Sigma$.

$$\begin{array}{llll}
(\mathrm{W}) & \dfrac{\Gamma \vdash^\Sigma M : A \quad x\notin\Gamma}{\Gamma, x{:}B \vdash^\Sigma M : A} & (\mathrm{C}) & \dfrac{\Gamma, x{:}B \vdash^\Sigma M : A \quad \Gamma \vdash^\Sigma N : B}{\Gamma \vdash^\Sigma M[x := N] : A} \\
(\mathrm{S}) & \dfrac{\Gamma, x{:}B \vdash^\Sigma M : A \quad x\notin\mathrm{FV}(M)}{\Gamma \vdash^\Sigma M : A} & (\leq \mathrm{L}) & \dfrac{\Gamma, x{:}B \vdash^\Sigma M : A \quad C \leq B}{\Gamma, x{:}C \vdash^\Sigma M : A}
\end{array}$$

As usual a generation lemma is handy: its proof can be found in [4].

**Lemma 1 (Generation lemma).**
*Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder and $\mathbb{T} = \mathbb{T}(\mathbb{C})$.*
(i) *Assume $A \not\sim \Omega$. Then $\Gamma \vdash^\Sigma x : A$ iff $(x{:}B)\in\Gamma$ and $B \leq A$ for some $B\in\mathbb{T}$.*
(ii) *Assume $A \not\sim \Omega$. Then $\Gamma \vdash^\Sigma MN : A$ iff $\Gamma \vdash^\Sigma M : B_i \to C_i$, $\Gamma \vdash^\Sigma N : B_i$, and $\bigcap_{i\in I} C_i \leq A$ for some $I$ and $B_i, C_i\in\mathbb{T}$.*
(iii) *Assume $\nu \not\leq A$. Then $\Gamma \vdash^\Sigma \lambda x.M : A$ iff $\Gamma, x{:}B_i \vdash^\Sigma M : C_i$, and $\bigcap_{i\in I}(B_i \to C_i) \leq A$ for some $I$ and $B_i, C_i\in\mathbb{T}$.*

## 3. Filter $\lambda$-structures and (restricted) filter models

It is possible to use intersection types for building models for $\lambda$-calculus and some of its restrictions. Let us first recall the general notion of restricted $\lambda$-calculus.

**Definition 9 (Restricted $\lambda$-calculus).** *Let $\mathbb{R} \subseteq \{\langle(\lambda x.M)N, M[x := N]\rangle \mid M, N\in\Lambda\}$. The* restricted $\lambda$-calculus *$\lambda_{\mathbb{R}}$ is the calculus obtained from the standard $\lambda$-calculus by restricting the $\beta$-rule to the redexes in $\mathbb{R}$ (called $\beta$-$\mathbb{R}$-redexes).*

Next definition of (*restricted*) model is a generalization of the classical notion of model for the untyped $\lambda$-calculus of Hindley-Longo (see [16]).

**Definition 10 ((Restricted) models).** *A* model for the (restricted) $\lambda$-calculus $\lambda_{\mathbb{R}}$ consists of a triple $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ such that $\mathcal{D}$ is a set, $\cdot : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$, *Env* $: Var \to \mathcal{V}$ for some $\mathcal{V} \subseteq \mathcal{D}$ and the interpretation function $[\![\ ]\!]^{\mathcal{D}} : \Lambda \times \mathit{Env} \to \mathcal{D}$ satisfies:

(i) $[\![x]\!]^{\mathcal{D}}_{\rho} = \rho(x)$;

(ii) $[\![MN]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho} \cdot [\![N]\!]^{\mathcal{D}}_{\rho}$;

(iii) $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} \cdot [\![N]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho[x:=[\![N]\!]^{\mathcal{D}}_{\rho}]}$ *for* $\langle (\lambda x.M)N, M[x := N] \rangle \in \mathbb{R}$;

(iv) *If* $\rho(x) = \rho'(x)$ *for all* $x \in \mathrm{FV}(M)$, *then* $[\![M]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho'}$;

(v) *If* $y \notin \mathrm{FV}(M)$, *then* $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda y.M[x := y]]\!]^{\mathcal{D}}_{\rho}$;

(vi) *If* $\forall d \in \mathcal{D}.[\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} = [\![N]\!]^{\mathcal{D}}_{\rho[x:=d]}$, *then* $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\rho}$.

$\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ *is* extensional *if moreover*

$$[\![\lambda x.Mx]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho}.$$

We can devise semantics domains out of intersection types by means of an appropriate notion of filter over a type preorder. This is a particular case of filter over a generic $\top$-meet semi-lattice (see [19]).

**Definition 11 ($\Sigma$-filters).** *Let* $\Sigma = (\mathbb{C}, \leq)$ *be a type preorder and* $\mathbb{T} = \mathbb{T}(\mathbb{C})$. *A* $\Sigma$*-filter (or a filter over* $\mathbb{T}$*) is a set* $X \subseteq \mathbb{T}$ *such that*

(i) *if* $\Omega \in \mathbb{C}$ *then* $\Omega \in X$;

(ii) *if* $A \leq B$ *and* $A \in X$, *then* $B \in X$;

(iii) *if* $A, B \in X$, *then* $A \cap B \in X$.

$\mathcal{F}^{\Sigma}$ *denotes the set of* $\Sigma$*-filters.*

NOTATION. Given $X \subseteq \mathbb{T}$, $\uparrow X$ denotes the $\Sigma$-filter generated by $X$. For $A \in \mathbb{T}$, we write $\uparrow A$ instead of $\uparrow \{A\}$.

It is possible to turn the space of filters into an applicative structure in which to interpret $\lambda$-terms. Assuming the Stone duality viewpoint, the interpretation of terms coincides with the sets of types which are deducible for them.

**Definition 12 (Filter structures).**

(i) *Application* $\_ \cdot \_ : \mathcal{F}^{\Sigma} \times \mathcal{F}^{\Sigma} \to \mathcal{F}^{\Sigma}$ *is defined as*

$$X \cdot Y = \uparrow \{B \mid \exists A \in Y. A \to B \in X\}.$$

(ii) *For any* $\lambda$*-term* $M$ *and environment* $\rho : Var \to \mathcal{F}^{\Sigma} \setminus \{\emptyset\}$,

$$[\![M]\!]^{\Sigma}_{\rho} = \{A \mid \exists \Gamma \models \rho.\ \Gamma \vdash^{\Sigma} M : A\},$$

*where* $\Gamma \models \rho$ *if and only if* $(x : B) \in \Gamma$ *implies* $B \in \rho(x)$.

(iii) *A* filter $\lambda$-structure *is a triple* $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma} \rangle$.

By rules ($\Omega$), ($\leq$) and ($\cap$I) the interpretations of all $\lambda$-terms are filters.

Thanks to the following theorem, it is sufficient that clause (iii) of Definition 10 holds in order a filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma} \rangle$ be also a model for the restricted $\lambda$-calculus $\lambda_{\mathbb{R}}$ (called *filter model for* $\lambda_{\mathbb{R}}$).

**Theorem 1.** *For all type preorders $\Sigma$ the interpretation function $\llbracket\ \rrbracket^{\Sigma}$ satisfies conditions (i), (ii), (iv), (v), (vi) of Definition 10.*

PROOF. We only consider the interesting cases.

(ii) Let $A{\in}\llbracket MN\rrbracket^{\Sigma}_{\rho}$. The case $A \sim \Omega$ is trivial. Otherwise there exists $\Gamma \models \rho$ such that $\Gamma \vdash^{\Sigma} MN : A$. By Lemma 1(ii) there exist $I$ and $B_i, C_i{\in}\mathbb{T}$ such that for all $i{\in}I$, $\Gamma \vdash^{\Sigma} M : B_i \to C_i$, $\Gamma \vdash^{\Sigma} N : B_i$, and $\bigcap_{i\in I} C_i \leq A$. From the first two judgments above, we get $B_i{\in}\llbracket N\rrbracket^{\Sigma}_{\rho}$ and $B_i \to C_i{\in}\llbracket M\rrbracket^{\Sigma}_{\rho}$. By definition of application it follows $A{\in}\llbracket M\rrbracket^{\Sigma}_{\rho} \cdot \llbracket N\rrbracket^{\Sigma}_{\rho}$.
Let now $A{\in}\llbracket M\rrbracket^{\Sigma}_{\rho} \cdot \llbracket N\rrbracket^{\Sigma}_{\rho}$. Then there exist $I$, $B_i, C_i{\in}\mathbb{T}$ such that $\bigcap_{i\in I} C_i \leq A$ and for any $i{\in}I$, $B_i \to C_i{\in}\llbracket M\rrbracket^{\Sigma}_{\rho}$ and $B_i{\in}\llbracket N\rrbracket^{\Sigma}_{\rho}$, hence there exist two bases over $\mathbb{C}$, $\Gamma_i$ and $\Gamma'_i$, such that $\Gamma_i \models \rho$, $\Gamma'_i \models \rho$, and moreover $\Gamma_i \vdash^{\Sigma} M : B_i \to C_i$, $\Gamma'_i \vdash^{\Sigma} N : B_i$. Consider the basis $\Gamma^* = \uplus_{i\in I}(\Gamma_i \uplus \Gamma'_i)$. We have $\Gamma^* \models \rho$, $\Gamma^* \vdash^{\Sigma} M : B_i \to C_i$ and $\Gamma^* \vdash^{\Sigma} N : B_i$. From the last two judgments, by applying ($\to$E), we deduce $\Gamma^* \vdash^{\Sigma} MN : C_i$, which implies, by ($\cap$I) and ($\leq$), $\Gamma^* \vdash^{\Sigma} MN : A$, hence $A{\in}\llbracket MN\rrbracket^{\Sigma}_{\rho}$.

(vi) Suppose that the premise holds and $A{\in}\llbracket \lambda x.M\rrbracket^{\Sigma}_{\rho}$. The case $\nu \leq A$ is trivial. Otherwise there is $\Gamma \models \rho$ such that $\Gamma \vdash^{\Sigma} \lambda x.M : A$. Since $x{\notin}\mathrm{FV}(\lambda x.M)$ by rule (S) we can assume $x{\notin}\Gamma$. By Lemma 1(iii) there exist $I$ and $B_i, C_i{\in}\mathbb{T}$ such that, for each $i{\in}I$, $\Gamma, x : B_i \vdash^{\Sigma} M : C_i$, and $\bigcap_{i\in I}(B_i \to C_i) \leq A$. By the premise, we get, for each $i{\in}I$, $\Gamma, x : B_i \vdash^{\Sigma} N : C_i$, hence by ($\to$I) and ($\leq$) we get $\Gamma \vdash^{\Sigma} \lambda x.N : A$, which implies $\llbracket \lambda x.M\rrbracket^{\Sigma}_{\rho} \subseteq \llbracket \lambda x.N\rrbracket^{\Sigma}_{\rho}$. Similarly one proves $\llbracket \lambda x.N\rrbracket^{\Sigma}_{\rho} \subseteq \llbracket \lambda x.M\rrbracket^{\Sigma}_{\rho}$.

**Corollary 1 ((Restricted) filter models).** *A filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, \llbracket\ \rrbracket^{\Sigma}\rangle$ is a* filter model *for the restricted $\lambda$-calculus $\lambda_{\mathbb{R}}$ iff for any redex $(\lambda x.M)N{\in}\mathbb{R}$, environment $\rho$,*

$$\llbracket(\lambda x.M)N\rrbracket^{\Sigma}_{\rho} = \llbracket M\rrbracket^{\Sigma}_{\rho[x:=\llbracket N\rrbracket^{\Sigma}_{\rho}]}\ \text{, that is,}$$

$$(\natural) \qquad \exists \Gamma \models \rho.\ \Gamma \vdash^{\Sigma} (\lambda x.M)N : A \quad \Leftrightarrow \quad \exists \Gamma' \models \rho.\ \Gamma' \vdash^{\Sigma} M[x := N] : A.$$

## 4. Four characterization results

The first two characterization results we give concern natural type preorders. We begin studying the representability of interesting classes of (strict) Scott continuous functions. These characterizations generalise those given in [8].

**Definition 13 (Representable functions).** *Given a type preorder $\Sigma$, a function $f : \mathcal{F}^{\Sigma} {\to} \mathcal{F}^{\Sigma}$ is said to be* representable in $\Sigma$ *if it is representable in the induced filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, \llbracket\ \rrbracket^{\Sigma}\rangle$, that is there exists $X{\in}\mathcal{F}^{\Sigma}$ such that for any $Y{\in}\mathcal{F}^{\Sigma}$, $X \cdot Y = f(Y)$.*

Next lemma is useful for characterizing the sets of representable functions.

**Lemma 2.** *Let $\Sigma = (\mathbb{C}, \leq)$ be a natural type preorder and $f : \mathcal{F}^{\Sigma} \to \mathcal{F}^{\Sigma}$ be a continuous function. Then $f$ is representable iff for all $I$ and $A_i, B_i, C, D{\in}\mathbb{T}(\mathbb{C})$, with $D \not\sim \Omega$, it holds*

$$(\flat) \quad (\forall i{\in}I. B_i{\in}f(\uparrow A_i))\ \&\ \bigcap_{i\in I}(A_i \to B_i) \leq C \to D \quad \Rightarrow \quad D{\in}f(\uparrow C).$$

PROOF. (⇐) Let $X_f = \uparrow \{A \to B \mid B \in f(\uparrow A)\}$ We prove that for any $C \in \mathbb{T}(\mathbb{C})$ we have $f(\uparrow C) = X_f \cdot \uparrow C$.

$$\begin{aligned}
X \cdot \uparrow C &= \uparrow \{D \mid \exists C' \geq C.C' \to D \in X\} && \text{by definition of application}\\
&= \uparrow \{D \mid C \to D \in X\} && \text{by } (\eta)\\
&= \{D \mid C \to D \in X\} && \text{by } (\to\text{-}\cap)\\
&= \{D \mid \exists I, A_i, B_i.(\forall i \in I.B_i \in f(\uparrow A_i))\& \\
&\quad \textstyle\bigcap_{i \in I}(A_i \to B_i) \leq C \to D\} && \text{by definition of } X_f\\
&= \{D \mid D \in f(\uparrow C)\} && \text{by } (\flat)\\
&= f(\uparrow C).
\end{aligned}$$

(⇒) Suppose by contradiction that there exist $I, A_i, B_i, C, D$ (with $D \not\sim \Omega$), such that $\bigcap_{i \in I}(A_i \to B_i) \leq C \to D$, and for any $i \in I$, $B_i \in f(\uparrow A_i)$, but $D \notin f(\uparrow C)$. If $X$ is any filter candidate to represent $f$, then, for any $i \in I$, $B_i \in X \cdot \uparrow A_i$, which implies, by easy computations, $A_i \to B_i \in X$, for any $i \in I$. Since $\bigcap_{i \in I}(A_i \to B_i) \leq C \to D$, it follows $C \to D \in X$, hence $D \in X \cdot \uparrow C$, making it impossible that $X$ represents $f$.

**Theorem 2 (Characterization of sets of representable functions).**
(i) *The set of functions representable in a strict natural preorder $\Sigma = (\mathbb{C}, \leq)$ contains:*

*(1) the step function $\perp \Rightarrow \perp$;*

*(2) the strict step functions iff $A \to B \leq C \to D$ imply $C \leq A, B \leq D$ for all $A, B, C, D \in \mathbb{T}(\mathbb{C})$;*

*(3) the strict continuous functions iff $\Sigma$ is a beta preorder.*

(ii) *The set of functions representable in a natural preorder $\Sigma = (\mathbb{C}, \leq)$ contains:*

*(1) the step function $\perp \Rightarrow \perp$ iff $A \to B \sim \Omega$ implies $B \sim \Omega$;*

*(2) the constant functions iff $\Omega \to B \leq C \to D$ implies $B \leq D$ for all $B, C, D$ in $\mathbb{T}(\mathbb{C})$;*

*(3) the step functions iff $A \to B \leq C \to D$ and $D \not\sim \Omega$ imply $C \leq A, B \leq D$ for all $A, B, C, D \in \mathbb{T}(\mathbb{C})$;*

*(4) the continuous functions iff $\Sigma$ is a beta preorder.*

PROOF. (sketch) For each point above, the theses follow applying condition $(\flat)$ of Lemma 2 to the class of functions involved, taking into account that:

- $D \in (\uparrow A \Rightarrow \uparrow B)(\uparrow C)$ iff $C \leq A$ and $B \leq D$;
- $B \in f(\uparrow A)$ iff $\uparrow A \Rightarrow \uparrow B \sqsubseteq f$;

where $\uparrow A \Rightarrow \uparrow B$ is the step function from $\uparrow A$ to $\uparrow B$, $f$ is a continuous function and $\sqsubseteq$ is the point-wise ordering.

All the type theories of Figure 2 are beta. Moreover, the type preorders $\mathcal{AO}, \mathcal{BCD}$ ($\mathcal{CDV}, \mathcal{EHR}$) are (strict) natural and therefore, by Theorem 2(ii4), in all the filter $\lambda$-structures induced by such preorders all (strict) continuous functions are representable.

Our second characterization result on natural type preorders consists in giving a criterion for selecting those type preorders whose induced filter $\lambda$-structures are indeed filter models of the whole $\lambda$-calculus. To do that we use a result of [20], in which an applicative structure is showed to be a model provided that it contains the combinators $\mathbf{K}$, $\mathbf{S}$ and $\varepsilon$. Thus, a condition for having a filter model can be obtained by simply forcing the existence of such combinators.

**Theorem 3 (Characterization of model-inducing preorders).** *Let $\Sigma = (\mathbb{C}, \leq)$ be a natural type preorder. The filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma} \rangle$ is a filter model of the whole $\lambda$-calculus iff the following three conditions are fulfilled.*

(i) *(existence of $\mathbf{K}$)*
$\forall C, D, E\ \exists I, A_i, B_i. \bigcap_{i \in I}(A_i \to B_i \to A_i) \leq C \to D \to E\ \Leftrightarrow$
$C \leq E$;

(ii) *(existence of $\mathbf{S}$)*
$\forall D, E, F, G\ \exists I, A_i, B_i, C_i. \bigcap_{i \in I}((A_i \to B_i \to C_i) \to (A_i \to B_i) \to A_i \to C_i)$
$\leq D \to E \to F \to G\ \Leftrightarrow$
$\exists H. E \leq F \to H$ *and* $D \leq F \to H \to G$;

(iii) *(existence of $\varepsilon$)*
$\forall C, D\ \exists I, A_i, B_i. \bigcap_{i \in I}((A_i \to B_i) \to A_i \to B_i) \leq C \to D\ \Leftrightarrow$
$\exists J, E_j, F_j. C \leq \bigcap_{j \in J}(E_j \to F_j) \leq D$.

*The filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma} \rangle$ is an extensional model if the third condition above is replaced by:*
(iii′) $\forall A\ \exists I, A_i, B_i. A \sim \bigcap_{i \in I}(A_i \to B_i)$.

These conditions are obtained by considering the application of combinators to filters. Similarly, one could characterize the representability of an arbitrary combinator of the shape $\lambda x_1 \ldots x_n.\mathsf{C}$, where $\mathsf{C}$ is a combination of variable (that is, it does not contain any $\lambda$-abstraction).

Our third result characterizes those type preorders inducing filter models for the main restricted $\lambda$-calculi studied in the literature, namely the $\lambda\mathbf{I}$-calculus [11], the $\lambda\mathbf{K}\mathsf{N}$-calculus [17] and the call-by-value $\lambda$-calculus [21]. The redexes of these calculi are defined as follows.

**Definition 14 (Restricted redexes).** *([21, 11, 17])*

(i) *A redex $(\lambda x.M)N$ is a $\beta_v$-*redex *if $N$ is a variable or an abstraction.*

(ii) *A redex $(\lambda x.M)N$ is a $\beta$-$\mathbf{I}$-*redex *if $x \in \mathrm{FV}(M)$.*

(iii) *A redex $(\lambda x.M)N$ is a $\beta$-$\mathbf{K}\mathsf{N}$-*redex *if it is a $\beta$-$\mathbf{I}$-redex or $N$ is either a variable or a closed strongly normalising term.*

Before characterizing (restricted) filter models we need a technical result on typing properties of strongly normalizing terms: for a proof see [13].

**Proposition 1 (Characterization of strongly normalizing terms).** *A $\lambda$-term $M$ is strongly normalizing iff for any type preorder $\Sigma = (\mathbb{C}, \leq)$ there exists $A \in \mathbb{T}(\mathbb{C})$ and a basis $\Gamma$ over $\mathbb{C}$ such that $\Gamma \vdash^{\Sigma} M : A$.*

Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder: we say that a basis $\Gamma^*$ over $\mathbb{C}$ is a $\langle \Sigma, \Gamma, M \rangle$-basis iff the subjects of $\Gamma^*$ are the variables which occur free in $M$ and are not subjects of $\Gamma$, i.e. $\{x \in \Gamma^*\} = \{x \in \mathrm{FV}(M) \mid x \notin \Gamma\}$.

**Theorem 4 (Characterizations of (restricted) filter models).** *Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder and $\Im(\Sigma, M, x)$[2] be short for:*

$$\forall \Gamma \textit{basis over } \mathbb{C}\ \forall \Gamma^* \langle \Sigma, \Gamma, M\rangle\textit{-basis}\ \ \forall A, B {\in} \mathbb{T}(\mathbb{C}).$$
$$\Gamma \vdash^{\Sigma} \lambda x.M : B {\to} A \ \Rightarrow\ \Gamma^*, \Gamma, x{:}B \vdash^{\Sigma} M : A.$$

*The filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, \llbracket\ \rrbracket^{\Sigma}\rangle$:*

(i) *is a model of the call-by-value $\lambda$-calculus iff for any $M$, $x$, $\rho$:*
*(1)* $\llbracket \lambda x.M \rrbracket^{\Sigma}_{\rho} \neq \emptyset$ *and*
*(2)* $\Im(\Sigma, M, x)$ *holds;*
(ii) *is a model of the $\lambda$**I**-calculus iff for any $M$, $x$, $N$, $\rho$ such that $x{\in}\mathrm{FV}(M)$:*
*(1)* $\llbracket M[x := N] \rrbracket^{\Sigma}_{\rho} \neq \emptyset$ *implies* $\llbracket N \rrbracket^{\Sigma}_{\rho} \neq \emptyset$ *and*
*(2)* $\Im(\Sigma, M, x)$ *holds;*
(iii) *is a model of the $\lambda$**K**N-calculus iff it is a model of the $\lambda$**I**-calculus;*
(iv) *is a model of the whole $\lambda$-calculus iff for any $M$, $\rho$:*
*(1)* $\llbracket M \rrbracket^{\Sigma}_{\rho} \neq \emptyset$ *and*
*(2)* $\Im(\Sigma, M, x)$ *holds.*

PROOF. We prove with details point (i) and give hints for the other points.

($\Rightarrow$) Let $\langle \mathcal{F}^{\Sigma}, \cdot, \llbracket\ \rrbracket^{\Sigma}\rangle$ be a model of the call-by-value calculus. Assume by contradiction that $\llbracket \lambda x.M_0 \rrbracket^{\Sigma}_{\rho_0} = \emptyset$ for some $M_0$, $x$, $\rho_0$: this implies $\llbracket (\lambda zy.y)(\lambda x.M_0) \rrbracket^{\Sigma}_{\rho_0} = \llbracket \lambda zy.y \rrbracket^{\Sigma}_{\rho_0} \cdot \llbracket \lambda x.M_0 \rrbracket^{\Sigma}_{\rho_0} = \emptyset$. Since $(\lambda zy.y)(\lambda x.M_0)$ is a $\beta_v$-redex which reduces to $\lambda y.y$ and $A \to A {\in} \llbracket \lambda y.y \rrbracket^{\Sigma}_{\rho}$ for all $\rho$ this contradicts ($\Leftarrow$) of condition ($\natural$) in Corollary 1. Now we prove that $\Im(\Sigma, M, x)$ holds for any $M, x$. Let $\Gamma \vdash^{\Sigma} \lambda x.M : B \to A$. Then $\Gamma' \vdash^{\Sigma} (\lambda x.M)x : A$, where $\Gamma' = \Gamma, x{:}B$. Let $\rho$ be an environment such that $\rho(y) = \uparrow C$ if $y{:}C {\in} \Gamma'$. It is easy to check that $\Gamma' \models \rho$. Since $(\lambda x.M)x$ is a $\beta_v$-redex, condition ($\natural$) of Corollary 1 holds, hence there exists a basis $\Gamma''$ such that $\Gamma'' \models \rho$ and moreover $\Gamma'' \vdash^{\Sigma} M : A$. By definition of $\models$, we have that for any variable $y$, if $y{:}D {\in} \Gamma''$ and $y{:}C {\in} \Gamma'$ then $C \leq D$. Applying rules $(\mathrm{S})$ and $(\leq \mathrm{L})$ we obtain $\Gamma^*, \Gamma, x{:}B \vdash^{\Sigma} M : A$ where $\Gamma^* = \{z : E {\in} \Gamma'' \mid z {\in} \mathrm{FV}(M)\,\&\, z {\notin} \Gamma\}$ is a $\langle \Sigma, \Gamma, M\rangle$-basis. Notice that the predicates in $\Gamma^*$ must vary according to the environment $\rho$, and that $\rho$ by construction can assign arbitrary filters to the subjects of $\Gamma^*$. The proof of $(\Rightarrow)$ is so complete.

($\Leftarrow$) First we show ($\Rightarrow$) of condition ($\natural$) in Corollary 1. From $\Gamma \vdash^{\Sigma} (\lambda x.M)N : A$ by Lemma 1(ii) and $\Im(\Sigma, M, x)$ we get $\Gamma^*, \Gamma, x : B_i \vdash^{\Sigma} M : C_i$, $\Gamma \vdash^{\Sigma} N : B_i$, and $\bigcap_{i \in I} C_i \leq A$ for all $\langle \Sigma, \Gamma, M\rangle$-basis $\Gamma^*$ and for some $I$, $B_i, C_i {\in} \mathbb{T}$. Then $\Gamma^*, \Gamma \vdash M[x := N] : C_i$ follows by rules (C) and (W), and so $\Gamma^*, \Gamma \vdash^{\Sigma} M[x := N] : A$ using rules ($\cap$I) and ($\leq$). We conclude observing that we can choose $\Gamma^*$ such that $\Gamma^* \models \rho$.

As to ($\Leftarrow$) of ($\natural$) let D be a deduction of $\Gamma \vdash M[x := N] : A$ and $\Gamma_i \vdash N : B_i$ for $i {\in} I$ be all the statements in D whose subject is $N$. Without loss of generality we can assume that $x$ does not occur in $\Gamma$.

If $I$ is non-empty, notice that $\Gamma \subseteq \Gamma_i$ but $\Gamma \upharpoonright \mathrm{FV}(N) = \Gamma_i \upharpoonright \mathrm{FV}(N)$ (by $\Gamma \upharpoonright \mathcal{X}$ we denote $\{x : A {\in} \Gamma \mid x {\in} \mathcal{X}\}$). So using rules (S) and $(\cap\mathrm{I})$, we have that $\Gamma \vdash N : \bigcap_{i \in I} B_i$. Moreover, one can easily see, by induction on $M$, that $\Gamma, x : \bigcap_{i \in I} B_i \vdash M : A$. Thus, by rule $(\to\mathrm{I})$, we have $\Gamma \vdash \lambda x.M : \bigcap_{i \in I} B_i {\to} A$. Hence, by $(\to\mathrm{E})$ we can conclude $\Gamma \vdash (\lambda x.M)N : A$.

[2] Notice that $\langle \Sigma, \Gamma, M\rangle$-bases in $\Im(\Sigma, M, x)$ are useful only for $\lambda\cap^{\Sigma}_{\nu}$, since $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\Omega}$ enjoy the sub-formula property.

If $I$ is empty, we get from D a derivation of $\Gamma \vdash M : A$ by replacing each $N$ by $x$. Two cases have to be considered:

- if $N$ is a variable, say $x$, then by Definition 12(ii) $\rho(x) \neq \emptyset$;
- if $N$ is an abstraction, then $[\![N]\!]^{\Sigma}_{\rho} \neq \emptyset$ by hypothesis.

In both cases there is a basis $\Gamma' \models \rho$, such that $\Gamma' \vdash^{\Sigma}_{\nu} N : B$ for some type $B$. By rule (W) we get $\Gamma, x : B \vdash M : A$ and we can conclude $\Gamma \uplus \{x : B\} \uplus \Gamma' \vdash (\lambda x.M)N : A$.

As to the proofs of the other points, proceed as in the previous case taking into account for point (iii) that if $N$ is a closed strongly normalizing term, by Proposition 1 it is typable in all intersection type systems from the empty basis.

Notice that $\Omega \in \mathbb{C}$ or $\nu \in \mathbb{C}$ implies condition (i1) of Theorem 4, $\Omega \in \mathbb{C}$ or $\nu \notin \mathbb{C}$ implies condition (ii1) of Theorem 4 and $\Omega \in \mathbb{C}$ implies condition (iv1) of Theorem 4.

The characterization of filter models can be extended to encompass extensionality. To this aim it is useful to know when typing is invariant under $\eta$-expansion and $\eta$-reduction. Let

$$(\eta\text{-}exp)\quad \frac{M \to_{\eta} N \qquad \Gamma \vdash N : A}{\Gamma \vdash M : A} \qquad\qquad (\eta\text{-}red)\quad \frac{M \to_{\eta} N \qquad \Gamma \vdash M : A}{\Gamma \vdash N : A}$$

Next proposition corresponds to Theorem 4.5 of [3].

**Proposition 2 (Characterization of subject $\eta$-reduction/expansion).**

(i) *Rule $(\eta\text{-}exp)$ is admissible in $\lambda\cap^{\Sigma}$ iff $\Sigma$ is eta;*

(ii) *Rule $(\eta\text{-}red)$ is admissible in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ iff $\Sigma$ validates $\mathcal{CDV}$, in $\lambda\cap^{\Sigma}_{\Omega}$ iff $\Sigma$ validates $\mathcal{CDV} \cup \{(\Omega\text{-}\eta)\}$, and it is never admissible in $\lambda\cap^{\Sigma}_{\nu}$.*

**Theorem 5 (Characterization of extensional (restricted) filter models).**
*Let $\Sigma = (\mathbb{C}, \leq)$ be a type preorder. The filter $\lambda$-structure $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma} \rangle$ is a extensional filter model of the restricted $\lambda$-calculus $\lambda_{\mathbb{R}}$ iff it is a model of $\lambda_{\mathbb{R}}$, $\Sigma$ is an eta type preorder which validates $\mathcal{CDV}$, and moreover if $\Omega \in \mathbb{C}$ then $\Sigma$ validates axiom $(\Omega\text{-}\eta)$, if $\nu \in \mathbb{C}$ then $\nu \in [\![M]\!]^{\Sigma}_{\rho}$ for all $M, \rho$.*

PROOF. ($\Rightarrow$) Let $\varphi \in \mathbb{C}$ be a constant that does not satisfy all the conditions in Definition 4(ii). One can show that $\varphi \notin [\![\lambda y.xy]\!]^{\Sigma}_{\rho[x:=\uparrow\varphi]}$: this implies that $\Sigma$ must be eta.
We have $A \to B \cap C \in [\![\lambda y.xy]\!]^{\Sigma}_{\rho[x:=\uparrow(A\to B)\cap(A\to C)]}$ for all $A, B, C$, but $A \to B \cap C \in \uparrow (A \to B) \cap (A \to C)$ only if $\Sigma$ validates axiom ($\to$-$\cap$). Similarly one can show that $\Sigma$ must validate axiom $(\eta)$ and axiom $(\Omega\text{-}\eta)$ when $\Omega \in \mathbb{C}$. Lastly if $\nu \in \mathbb{C}$ then $\nu \in [\![\lambda x.Mx]\!]^{\Sigma}_{\rho}$ for all $M, \rho$ by axiom $(\text{Ax-}\nu)$ implies $\nu \in [\![M]\!]^{\Sigma}_{\rho}$ for all $M, \rho$.

($\Leftarrow$) follows from Proposition 2, but the case $\nu \in \mathbb{C}$, in which $\nu$ is harmless being contained in the interpretations of all terms.

Using the previous theorems we get: $\langle \mathcal{F}^{\nabla}, \cdot, [\![\ ]\!]^{\nabla} \rangle$ with $\nabla \in \{\mathcal{CDV}\}$ is a model of the $\lambda$**I**-calculus, with $\nabla = \mathcal{EHR}$ is a model of the call-by-value $\lambda$-calculus, with $\nabla \in \{\mathcal{AO}, \mathcal{BCD}\}$ is a model of the whole $\lambda$-calculus.

## 5. Trimmed filter models

In this section we provide actual examples of filter models of the whole $\lambda$-calculus where not all continuous functions are representable. In particular we devise two type preorders which are natural but not beta. This implies, by Theorem 2(ii4), that some continuous function cannot be represented in the filter $\lambda$-structures induced by such preorders. For the first model we acknowledge the adaptation of an idea in [8].

**Definition 15 (Trimmed models).** *A (strict) filter model will be called* trimmed *if not all the (strict) continuous functions are representable in it.*

### The type preorder $\Sigma^{\Diamond}$

**Definition 16 ($\Sigma^{\Diamond}$).** *Let* $\mathbb{C}^{\Diamond} = \{\Omega, \Diamond, \heartsuit\}$. *The type preorder* $\Sigma^{\Diamond}$ *is the preorder induced by the set of rules* $\Diamond = \mathcal{BCD} \cup \{(\Diamond\text{-}\heartsuit)\}$*, where*

$$(\Diamond\text{-}\heartsuit)\ A \leq A[\Diamond := \heartsuit].$$

In order to show that $\Sigma^{\Diamond}$ induces a trimmed filter model we need a few technical results.

**Lemma 3.** (i) $A \leq_{\Diamond} B$ *implies* $A[\Diamond := \heartsuit] \leq_{\Diamond} B[\Diamond := \heartsuit]$*;*

(ii) $\Gamma \vdash^{\Diamond} M : A$ *implies* $\Gamma[\Diamond := \heartsuit] \vdash^{\Diamond} M : A[\Diamond := \heartsuit]$*;*

(iii) $\Gamma, \Gamma' \vdash^{\Diamond} M : A$ *implies* $\Gamma, \Gamma'[\Diamond := \heartsuit] \vdash^{\Diamond} M : A[\Diamond := \heartsuit]$*;*

(iv) $\forall i{\in}I.\ \Gamma, x : A_i \vdash^{\Diamond} M : B_i$ *and* $\bigcap_{i\in I}(A_i{\to}B_i) \leq_{\Diamond} \bigcap_{j\in J}(C_j{\to}D_j)$ *imply* $\forall j{\in}J.\ \Gamma, x : C_j \vdash^{\Diamond} M : D_j$.

PROOF. (i) By induction on the definition of $\leq_{\Diamond}$.

(ii) By induction on derivations using (i) for rule $(\leq_{\Diamond})$.

(iii) From (ii) and the admissible rule ($\leq_{\Diamond}$L), taking into account that if $(x : B){\in}\Gamma$, then $(x : B[\Diamond := \heartsuit]){\in}\Gamma[\Diamond := \heartsuit]$ and $B \leq_{\Diamond} B[\Diamond := \heartsuit]$.

(iv) We shall denote by $\psi, \xi$ (possibly with indexes) elements of $\mathbb{C}^{\Diamond}$.We show by induction on the definition of $\leq_{\Diamond}$ that

$$(\textstyle\bigcap_{i\in I}(A_i{\to}B_i)) \cap (\bigcap_{h\in H}\psi_h) \leq_{\Diamond} (\bigcap_{j\in J}(C_j{\to}D_j)) \cap (\bigcap_{k\in K}\xi_k)$$
$$\text{and } \forall i{\in}I.\ \Gamma, x : A_i \vdash^{\Diamond} M : B_i \text{ imply } \forall j{\in}J.\ \Gamma, x : C_j \vdash^{\Diamond} M : D_j.$$

The only interesting case is when the applied rule is $(\Diamond\text{-}\heartsuit)$, i.e. we have

$$\bigcap_{i\in I}(A_i{\to}B_i) \cap (\bigcap_{h\in H}\psi_h) \leq_{\Diamond} (\bigcap_{i\in I}(A_i{\to}B_i) \cap (\bigcap_{h\in H}\psi_h))[\Diamond := \heartsuit].$$

By hypothesis $\Gamma, x : A_i \vdash^{\Diamond} M : B_i$, so we are done by (iii).

**Theorem 6.** *The natural type preorder* $\Sigma^{\Diamond}$ *is not beta, but* $\langle \mathcal{F}^{\Diamond}, \cdot, \llbracket\ \rrbracket^{\Diamond} \rangle$ *is a filter model of the whole $\lambda$-calculus.*

PROOF. A counter-example to the condition of Definition 4(i) is $\Diamond{\to}\Diamond \leq_{\Diamond} \heartsuit{\to}\heartsuit$, since $\heartsuit {\not\leq}_{\Diamond} \Diamond$.
To show that $\langle \mathcal{F}^{\Diamond}, \cdot, \llbracket\ \rrbracket^{\Diamond} \rangle$ is a model of the whole $\lambda$-calculus, it suffices, by Theorem 4(iv), to verify that $\Im(\Sigma^{\Diamond}, M, x)$ holds for any $M, x$. By Lemma 1(iii) $\Gamma \vdash^{\Diamond} \lambda x.M : A{\to}B$ implies $\Gamma, x : C_i \vdash^{\Diamond} M : D_i$ for some $I, C_i, D_i$ such that $\bigcap_{i\in I}(C_i{\to}D_i) \leq_{\Diamond} A{\to}B$. So, we are done by Lemma 3(iv).

The step function $\uparrow\Diamond \Rightarrow \uparrow\Diamond$ is an instance of non-representable function in $\mathcal{F}^{\Diamond}$.

**The type preorder $\varSigma^{\spadesuit}$**

**Definition 17 (The mapping $\mathsf{p}$).** *Let $\mathbb{C}^{\spadesuit} = \{\varOmega, \spadesuit, \clubsuit\}$ The mapping $\mathsf{p} : \mathbb{T}^{\spadesuit} \to \mathbb{T}^{\spadesuit}$ is inductively defined as follows:*

$$\begin{array}{l} \mathsf{p}(\varOmega) = \varOmega; \quad \mathsf{p}(\spadesuit) = \varOmega; \quad \mathsf{p}(\clubsuit) = \clubsuit; \\ \mathsf{p}(A \to B) = A \to \mathsf{p}(B); \\ \mathsf{p}(A \cap B) = \mathsf{p}(A) \cap \mathsf{p}(B). \end{array}$$

**Definition 18 ($\varSigma^{\spadesuit}$).** *$\varSigma^{\spadesuit}$ is the type preorder induced by the set of rules $\spadesuit = \mathcal{BCD} \cup \{(\spadesuit\text{-}\clubsuit), (\spadesuit\text{-}\clubsuit\text{-}\to)\}$ where:*

$$\begin{array}{ll} (\spadesuit\text{-}\clubsuit) & A \leq \mathsf{p}(A); \\ (\spadesuit\text{-}\clubsuit\text{-}\to) & A \to B \leq \mathsf{p}(A) \to \mathsf{p}(B). \end{array}$$

Given a basis $\varGamma$, let $\mathsf{p}(\varGamma)$ be the basis obtained by substituting any judgment $x{:}A$ of $\varGamma$ by $x{:}\mathsf{p}(A)$.

We show that $\varSigma^{\spadesuit}$ induces a trimmed filter model. As in the case of $\varSigma^{\diamondsuit}$, we prove some technical results in order to show that $\Im(\varSigma^{\spadesuit}, M, x))$ holds for any $M$, $x$.

**Lemma 4.** (i) *The mapping $\mathsf{p}$ is idempotent, i.e. $\mathsf{p}(\mathsf{p}(A)) = \mathsf{p}(A)$.*

(ii) *$A \to \mathsf{p}(B) \sim_{\spadesuit} \mathsf{p}(A) \to \mathsf{p}(B)$;*

(iii) *$A \leq_{\spadesuit} B$ implies $\mathsf{p}(A) \leq_{\spadesuit} \mathsf{p}(B)$;*

(iv) *$\varGamma \vdash^{\spadesuit} M : A$ implies $\mathsf{p}(\varGamma) \vdash^{\spadesuit} M : \mathsf{p}(A)$;*

(v) *$\varGamma, \varGamma' \vdash^{\spadesuit} M : A$ implies $\varGamma, \mathsf{p}(\varGamma') \vdash^{\spadesuit} M : \mathsf{p}(A)$;*

(vi) *$\forall i{\in}I.\ \varGamma, x : A_i \vdash^{\spadesuit} M : B_i$ and $\bigcap_{i \in I}(A_i \to B_i) \leq_{\spadesuit} \bigcap_{j \in J}(C_j \to D_j)$ imply $\forall j{\in}J.\ \varGamma, x : C_j \vdash^{\spadesuit} M : D_j$.*

PROOF. (i) Easy.

(ii) We get $\mathsf{p}(A) \to \mathsf{p}(B) \leq_{\spadesuit} A \to \mathsf{p}(B)$ by axiom $(\spadesuit\text{-}\clubsuit)$ and rule $(\eta)$. Moreover $A \to \mathsf{p}(B) \leq_{\spadesuit} \mathsf{p}(A) \to \mathsf{p}(\mathsf{p}(B)) = \mathsf{p}(A) \to \mathsf{p}(B)$ by axiom $(\spadesuit\text{-}\clubsuit\text{-}\to)$ and (i).

(iii) By induction on the definition of $\leq_{\spadesuit}$ using (i) and (ii).

(iv) By induction on derivations using (iii) for rule $(\leq_{\spadesuit})$. We give the details just for rule $(\to \mathrm{E})$. Suppose $M \equiv NL$ and $\varGamma \vdash^{\spadesuit} N : B \to A$, $\varGamma \vdash^{\spadesuit} L : B$. Then, by induction, $\mathsf{p}(\varGamma) \vdash^{\spadesuit} N : \mathsf{p}(B \to A)$, $\mathsf{p}(\varGamma) \vdash^{\spadesuit} L : \mathsf{p}(B)$. Since $\mathsf{p}(B \to A) = B \to \mathsf{p}(A)$, by (ii) we get $\mathsf{p}(\varGamma) \vdash^{\spadesuit} NL : \mathsf{p}(A)$.

(v) From (iv) and the admissible rule ($\leq_{\spadesuit}$ L), taking into account that if $(x : B) \in \varGamma$, then $(x : \mathsf{p}(B)) \in \mathsf{p}(\varGamma)$ and $B \leq_{\spadesuit} \mathsf{p}(B)$.

(vi) We shall denote by $\psi, \xi$ (possibly with indexes) elements of $\mathbb{C}^{\spadesuit}$.We show by induction on the definition of $\leq_{\spadesuit}$ that

$$(\textstyle\bigcap_{i \in I}(A_i \to B_i)) \cap (\bigcap_{h \in H} \psi_h) \leq_{\spadesuit} (\bigcap_{j \in J}(C_j \to D_j)) \cap (\bigcap_{k \in K} \xi_k)$$

$$\text{and } \forall i{\in}I.\ \varGamma, x : A_i \vdash^{\spadesuit} M : B_i \text{ imply } \forall j{\in}J.\ \varGamma, x : C_j \vdash^{\spadesuit} M : D_j.$$

The only interesting case is when the applied rules are $(\spadesuit\text{-}\clubsuit)$ or $(\spadesuit\text{-}\clubsuit\text{-}\to)$, i.e. we have $A \to B \leq_{\spadesuit} \mathsf{p}(A \to B) = A \to \mathsf{p}(B)$ or $A \to B \leq_{\spadesuit} \mathsf{p}(A) \to \mathsf{p}(B)$. By hypothesis $\varGamma, x{:}A \vdash^{\spadesuit} M : B$, so we are done by (v).

**Theorem 7.** *The natural type preorder $\Sigma^{\spadesuit}$ is not beta, but $\langle \mathcal{F}^{\spadesuit}, \cdot, [\![\ ]\!]^{\spadesuit} \rangle$ is a filter model of the whole $\lambda$-calculus.*

PROOF. As in the proof of Theorem 6, Lemma 4(vi) allows to prove that for any $M$, $x$, $\Im(\Sigma^{\spadesuit}, M, x)$ holds, hence we can apply Theorem 4(iv) in order to conclude that $\langle \mathcal{F}^{\spadesuit}, \cdot, [\![\ ]\!]^{\spadesuit} \rangle$ is a model. On the other hand $\spadesuit$ is not a beta theory. For instance, $\varphi{\to}\omega \leq_{\spadesuit} \Omega{\to}\omega$, but $\Omega \not\leq_{\spadesuit} \varphi$.

The step function $\uparrow\spadesuit \Rightarrow \uparrow\clubsuit$ is an example of function not representable in $\mathcal{F}^{\spadesuit}$.

Actually $\mathcal{F}^{\spadesuit}$ is the inverse limit solution of the domain equation $\mathcal{D} \simeq [\mathcal{D}{\to}\mathcal{D}]$ computed in the category of $p$-*lattices* (see [2]), whose objects are $\omega$-algebraic lattices $\mathcal{D}$ endowed with a finitary additive projection $\delta : \mathcal{D}{\to}\mathcal{D}$ and whose morphisms $f : (\mathcal{D}, \delta){\to}(\mathcal{D}', \delta')$ are continuous functions such that $\delta' \circ f \sqsubseteq f \circ \delta$.

## 6. Conclusions

When stepping into the world of $\lambda$-calculus semantics, intersection type systems turn out to be a useful "vehicle" to move around, since they provide a finitary way to describe and analyse particular classes of models. By simply adding a single constant or condition on a type preorder, a different semantical domain is characterized. One is then naturally induced to expect that intersection types will provide, in the long run, a sort of tailor shop in which particular domains can be tailored for any specific need. As a matter of fact, the possibility of pacing along this direction has been shown to be real also in [3].

In the present paper we have made a step forward in this direction. Filter $\lambda$-structures induced by intersection type preorders have been shown to provide models for the whole $\lambda$-calculus and for a number of relevant "restricted" $\lambda$-calculi when particular conditions on the type preorders are fulfilled. Even more, our proposed conditions provide precise characterizations for intersection type-induced models.

When a model is produced, the second step is almost always to make it precisely fit the calculus, by "trimming" it and eliminating the exceeding parts. We have shown in the present paper that in the framework of intersection-induced models for the $\lambda$-calculus such a trimming is indeed possible, by providing two examples of filter models in which not all the continuous functions are representable.

Much to do is left about model "tailoring", like trying to see if many conditions on type preorders implicitly expressed in terms of generation properties of type assignment can be made explicit on the type preorders itself. Besides it would be interesting to check whether also the webbed models [7] allow for "tailoring operations".

### Acknowledgments

The authors are grateful to Furio Honsell, Henk Barendregt and Wil Dekkers for enlightening discussions on the subject of the present paper. We wish also to thank the anonymous referees who pointed us interesting directions for further research and whose comments and suggestions have been very helpful to improve the presentation of the paper.

## References

1. Samson Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Inform. and Comput.*, 105(2):159–267, 1993.

2. Fabio Alessi. The category $p$-sfp. Internal Report n. 27/96/RR, Dipartimento di Matematica e Informatica, University of Udine, 1996.
3. Fabio Alessi, Franco Barbanera, and Mariangiola Dezani-Ciancaglini. Intersection types and computational rules. In Ruy de Queiroz, Elaine Pimentel, and Lucilia Figueiredo, editors, *WoLLIC'03*, volume 84 of *El. Notes in Theoret. Comput. Sci.* Elsevier, 2003.
4. Fabio Alessi, Mariangiola Dezani-Ciancaglini, and Furio Honsell. A complete characterization of complete intersection-type preorders. *ACM Trans. on Comput. Logic*, 4(1):120–147, 2003.
5. Fabio Alessi, Mariangiola Dezani-Ciancaglini, and Stefania Lusin. Intersection types and domain operators. *Theoret. Comput. Sci.*, 2004. to appear.
6. Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *J. Symbolic Logic*, 48(4):931–940 (1984), 1983.
7. Chantal Berline. From computation to foundations via functions and application: The $\lambda$-calculus and its webbed models. *Theoret. Comput. Sci.*, 249:81–161, 2000.
8. Mario Coppo, Mariangiola Dezani-Ciancaglini, Furio Honsell, and Giuseppe Longo. Extended type structures and filter lambda models. In Gabriele Lolli, Giuseppe Longo, and Anna Lisa Marcja, editors, *Logic colloquium '82*, pages 241–262. North-Holland, Amsterdam, 1984.
9. Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Functional characters of solvable terms. *Z. Math. Logik Grundlag. Math.*, 27(1):45–58, 1981.
10. Mario Coppo, Mariangiola Dezani-Ciancaglini, and Maddalena Zacchi. Type theories, normal forms, and $D_\infty$-lambda-models. *Inform. and Comput.*, 72(2):85–116, 1987.
11. Haskell B. Curry and Robert Feys. *Combinatory Logic*, volume I of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1958.
12. Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, and Silvia Likavec. Behavioural inverse limit models. *Theoret. Comput. Sci.*, 2004. to appear.
13. Mariangiola Dezani-Ciancaglini, Furio Honsell, and Yoko Motohama. Compositional characterization of $\lambda$-terms using intersection types. *Theoret. Comput. Sci.*, 2004. to appear.
14. Lavinia Egidi, Furio Honsell, and Simona Ronchi Della Rocca. Operational, denotational and logical descriptions: a case study. *Fund. Inform.*, 16(2):149–169, 1992.
15. Gerhard K. Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael W. Mislove, and Dana S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, 1980.
16. Roger Hindley and Giuseppe Longo. Lambda-calculus models and extensionality. *Z. Math. Logik Grundlag. Math.*, 26(4):289–310, 1980.
17. Furio Honsell and Marina Lenisa. Semantical analysis of perpetual strategies in $\lambda$-calculus. *Theoret. Comput. Sci.*, 212(1-2):183–209, 1999.
18. Furio Honsell and Simona Ronchi Della Rocca. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus. *J. Comput. System Sci.*, 45(1):49–75, 1992.
19. Peter T. Johnstone. *Stone spaces*. Cambridge University Press, Cambridge, 1986. Reprint of the 1982 edition.
20. Albert R. Meyer. What is a model of the lambda calculus? *Inform. and Control*, 52(1):87–122, 1982.
21. Gordon D. Plotkin. Call-by-name, call-by-value and the $\lambda$-calculus. *Theoret. Comput. Sci.*, 1(2):125–159, 1975.
22. Gordon D. Plotkin. Set-theoretical and other elementary models of the $\lambda$-calculus. *Theoret. Comput. Sci.*, 121(1-2):351–409, 1993.
23. Dana S. Scott. Open problem. In Corrado Böhm, editor, *Lambda Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, page 369. Springer-Verlag, Berlin, 1975.