Behavioural Inverse Limit λ-Models

Mariangiola Dezani-Ciancaglini [a,1], Silvia Ghilezan [b,2], Silvia Likavec [a,b,c,1]

[a] *Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy* {dezani,likavec}@di.unito.it

[b] *Faculty of Engineering, University of Novi Sad, Trg D. Obradovića 6, 21000 Novi Sad, Yugoslavia* gsilvia@uns.ns.ac.yu

[c] *École Normale Supérieure, 46, allé d'Italie, 69364 Lyon 07, France*

Abstract

We construct two inverse limit λ-models which completely characterise sets of terms with similar computational behaviours: the sets of normalising, head normalising, weak head normalising λ-terms, those corresponding to the persistent versions of these notions, and the sets of closable, closable normalising, and closable head normalising λ-terms. More precisely, for each of these sets of terms there is a corresponding element in at least one of the two models such that a term belongs to the set if and only if its interpretation (in a suitable environment) is greater than or equal to that element. We use the finitary logical description of the models, obtained by defining suitable intersection type assignment systems, to prove this.

Key words: Lambda calculus, intersection types, models of lambda calculus, Stone dualities, reducibility method.

1 Introduction

The aim of this paper is to present two λ-models which completely characterise well-known computational properties of λ-terms. We consider nine computational

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

[2] Partially supported by grant 1630 "Representation of proofs with applications, classification of structures and infinite combinatorics" (of the Ministry of Science, Technology, and Development of Serbia).

properties of λ-terms and corresponding nine sets of λ-terms: the sets of *normalising*, *head normalising*, *weak head normalising* λ-terms, those corresponding to the *persistent* versions of these notions, and the sets of *closable*, *closable normalising* and *closable head normalising* λ-terms.

We build two *inverse limit* λ*-models* $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$, according to Scott [28], which completely characterise each of the mentioned sets of terms. More precisely, for each of the above nine sets of terms there is a corresponding element in at least one of these models such that a term belongs to the set if and only if its interpretation (in a suitable environment) is greater than or equal to that element. This is proved by using the finitary logical descriptions of the models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$, obtained by defining two *intersection type assignment systems* in the following way. Starting from atomic types corresponding to the elements of $\mathcal{D}_0$ and $\mathcal{E}_0$, we construct the sets $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$ of types using the *function type* constructor and the *intersection type* constructor. Then, we define the sets $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{F}^{\mathcal{E}}$ of filters on the sets $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$, respectively. Following Scott [30], Coppo et al. [10], and Alessi [3], we will show that the sets $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{F}^{\mathcal{E}}$ (ordered by subset inclusion) and the corresponding inverse limit λ-models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are isomorphic as Scott domains. This isomorphism falls in the general framework of *Stone dualities* (Johnstone [18]). This framework later received a frame-theoretic explanation by Abramsky in the broader perspective of "domain theory in logical form" [1]. The interest of the above isomorphism lies in the fact that the interpretations of λ-terms in $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are isomorphic to the filters of types one can derive in the corresponding type assignment systems. This gives the desired finitary logical descriptions of the models. Therefore, the primary complete characterisation can be stated equivalently as follows: a term belongs to one of the nine sets mentioned if and only if it has a certain type (in a suitable basis) in one of the obtained type assignment systems.

In order to prove one part of this property we apply the *reducibility method*. It is a well-known method, based on a set-theoretic semantics of types, for proving the strong normalisation property of various type systems (Tait [32], Tait [33], Girard [17], Krivine [20], [21], Mitchell [24]). The reducibility method is also used in Leivant [22] and Gallier [14] for characterising strongly normalising terms, normalising terms, head normalising terms, and weak head normalising terms by their typeability in various intersection type systems. In Dezani et al. [13] the reducibility method is applied in order to characterise both the mentioned sets of terms and their persistent versions.

In all these papers different properties are characterised by means of different type assignment systems. So, the novelty of our approach is the characterisation of all nine computational properties of terms by means of *only two type assignment systems*, which induce λ-models. Moreover, in all the papers mentioned, different computational properties require different type interpretations in the reducibility method, whereas we adapt the reducibility method using *only two type interpretations* for all nine computational properties.

The most intriguing part of the other direction of the proof is the one concerning the persistently normalising terms. For that reason we develop a new characterisation of these terms which is less general but much simpler than the one presented in Dezani et al. [13].

Lastly, we remark that there are *apparently two* kinds of semantics for intersection types in the literature and that the present paper deals with both of them. The *set-theoretical* semantics, originally introduced in Barendregt et al. [6], generalises the one given by Scott for simple types (Scott [29]). The meanings of types are subsets of the domain of discourse, arrow types are defined as *logical predicates* and intersection is the set-theoretic intersection. This semantics is the basis of our application of the reducibility method. The second semantics views types as *compact elements* of Plotkin's λ-structures (Plotkin [26]). According to this interpretation, the universal type denotes the least element, intersections denote joins of compact elements, and arrow types allow to internalise the space of continuous endomorphisms. This semantics allows us to obtain the isomorphisms between the models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ and the sets $\mathcal{F}^\mathcal{D}$ and $\mathcal{F}^\mathcal{E}$ of filters of types. It is also true that these two kinds of semantics are strongly related. If $\mathcal{S}$ is a Scott domain, $\mathsf{c} \in \mathcal{S}$ is compact, and we define the basic open subset $\mathcal{O}_\mathsf{c} = \{\mathsf{d} \in \mathcal{S} \mid \mathsf{c} \sqsubseteq \mathsf{d}\}$, then the set of all $\mathcal{O}_\mathsf{c}$ is a basis for the topology on $\mathcal{S}$. Moreover, basic open subsets are closed under intersection and basic open subsets of function spaces are defined by logical relations.

The paper is organised as follows. Section 2 discusses the duality between intersection types and inverse limit domains. In Section 3 the models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are built. In Section 4 we define intersection types and build sets $\mathcal{F}^\mathcal{D}$ and $\mathcal{F}^\mathcal{E}$ of filters in order to prove the isomorphism between the inverse limit λ-models and the filter models. The corresponding intersection type assignment systems are defined in Section 5. The main result is a complete characterisation of computational behaviours of terms by their typeability in the corresponding type systems. This is stated in Section 5 and one direction of the equivalence is proved there for all cases but for persistently normalising terms. Section 6 deals with the case of persistently normalising terms, which needs the notions of replaceable and non-replaceable variables. In Section 7 we prove the other direction of the equivalence using standard techniques of the reducibility method adapted for these type systems.

A preliminary version of the present paper (dealing only with the first six sets of terms) was presented at the International Workshop on Rewriting in Proof and Computation (RPC'01, Tohoku University 25-27/10/2001, Sendai, Japan) [11] and at the Types Workshop (TYPES 2002 24-28/04/2002, Nijmegen, The Netherlands). An extended abstract of the present paper is [12].

2 Inverse Limit λ-Models and Intersection Types

Stone duality allows to describe special classes of topological spaces by means of (possibly finitary) partial orders. The seminal result is the duality between the categories of Stone spaces and of Boolean algebras (see Johnstone [18]). Other very important examples are the descriptions of ω-*algebraic complete lattices* as *intersection type theories* in Coppo et al. [10], *Scott domains* as *Scott information systems* in Scott [30], and *SFP domains* as *pre-locales* in Abramsky [1]. It is worthwhile to mention also Martin-Löf's domain interpretation of intuitionistic type theory in [23]. In fact, [23] gives an explicit syntactic presentation of domain theory and the discussions after Martin-Löf's presentation of this paper at the Chalmers Workshop raised a number of questions and conjectures mainly answered by Abramsky in [1]. Notice that all the above mentioned dualities are discussed in [18].

Intersection type theories offer a syntactic (i.e. finitary) approach to presenting the compact elements of certain domain constructions.

In the literature intersection types are usually employed for describing ω-algebraic complete lattices. Instead, in this paper we describe Scott domains (i.e. consistently complete ω-algebraic cpos) by means of intersection types, where the intersection between types is a partial operator. In fact intersection types with partial intersection and equipped with preorders (i.e. *intersection type theories*) become essentially equivalent to Scott information systems, as first stated in Coppo et al. [9] and proved in Alessi [3].

It is well known (see [28, 2]) that the category of Scott domains $\mathbf{DOM}$ with embedding projections has solutions for domain equations of the form

$$\begin{cases} \mathcal{X} \cong \mathbb{F}(\mathcal{X}) \\ \mathcal{C}_0 \longleftarrow \mathcal{X} \end{cases}$$

where $\mathbb{F} : \mathbf{DOM} \mapsto \mathbf{DOM}$ is continuous and $\mathcal{C}_0$ is a given Scott domain such that $\mathcal{C}_0 \longleftarrow \mathbb{F}(\mathcal{C}_0)$. The solution is obtained by computing the inverse limit of the chain of projections

$$\mathcal{C}_0 \longleftarrow \mathbb{F}(\mathcal{C}_0) \longleftarrow \mathbb{F}(\mathbb{F}(\mathcal{C}_0)) \ldots$$

We are interested in the following two cases:

- $\mathbb{F}$ is the function-space functor $\mathcal{X} \mapsto [\mathcal{X} \to \mathcal{X}]$,
- $\mathbb{F}$ is the lifted function-space functor $\mathcal{X} \mapsto [\mathcal{X} \to \mathcal{X}]_\perp$.

In each case, the solution $\mathcal{C}_\infty$ computed by the inverse limit produces a λ-model, called an *inverse limit λ-model* (see [7], Chapters 5 and 18).

More precisely, let $\mathbb{F}_\infty : \mathcal{C}_\infty \mapsto \mathbb{F}(\mathcal{C}_\infty)$ and $\mathbb{G}_\infty : \mathbb{F}(\mathcal{C}_\infty) \mapsto \mathcal{C}_\infty$ denote the two canonical isomorphisms of either solution, Env_∞ be the set of all mappings from the set of term variables to $\mathcal{C}_\infty$ and let ρ range over Env_∞. Then, the standard *semantic interpretation* $[\![\ ]\!]^\infty : \Lambda \times \mathsf{Env}_\infty \mapsto \mathcal{C}_\infty$ of λ-terms is given by:

$$\begin{aligned} [\![x]\!]^\infty_\rho &= \rho(x) \\ [\![\mathbf{tu}]\!]^\infty_\rho &= \mathbb{F}_\infty([\![\mathbf{t}]\!]^\infty_\rho)([\![\mathbf{u}]\!]^\infty_\rho) \\ [\![\lambda x.\mathbf{t}]\!]^\infty_\rho &= \mathbb{G}_\infty(\lambda\!\!\lambda \mathsf{d} \in \mathcal{C}_\infty.[\![\mathbf{t}]\!]^\infty_{\rho[\mathsf{d}/x]}), \end{aligned}$$

where for the third clause we have taken the advantage of the fact that, in any case, every continuous endofunction on $\mathcal{C}_\infty$ belongs to $\mathbb{F}(\mathcal{C}_\infty)$.

Clearly, the inverse limit λ-models can be described in terms of their compact elements. As stated in the proof sketch in Coppo et al. [10] and fully proved in Alessi [3], these can be denoted by taking:

(1) the types freely generated by closing (a set of atomic types corresponding to) the elements of the initial cpo under the *function type* constructor $\rightarrow$ and the *intersection type* constructor $\cap$ between *compatible* types, where two types are compatible if the corresponding elements have a join;
(2) the preorder between types induced by reversing the order in the initial cpo and by encoding the initial embedding, according to the correspondence:

function type constructor	$\rightsquigarrow$	step function
intersection type constructor	$\rightsquigarrow$	join.

To conclude, we recall a well-known relation between joins of step functions (see for example Gierz et al. [16]). Given compact elements a and b in the Scott domains $\mathcal{A}$ and $\mathcal{B}$ respectively, the step function $\mathsf{a} \Rightarrow \mathsf{b}$ is defined by

$$\lambda\!\!\lambda \mathsf{d}.\ \textbf{if } \mathsf{a} \sqsubseteq \mathsf{d} \textbf{ then } \mathsf{b} \textbf{ else } \bot.$$

Given a Scott continuous function $f : \mathcal{A} \mapsto \mathcal{B}$, one has that

f is greater than or equal to $\mathsf{a} \Rightarrow \mathsf{b}$ iff $f(\mathsf{a})$ is greater than or equal to b.

Since we wish to deal only with compact elements, we are interested in the case when $f : \mathcal{A} \mapsto \mathcal{B}$ is compact, i.e. $f = \bigsqcup_{i \in I}(\mathsf{a}_i \Rightarrow \mathsf{b}_i)$. Then we have:

$$\bigsqcup_{i \in I}(\mathsf{a}_i \Rightarrow \mathsf{b}_i) \sqsupseteq \mathsf{a} \Rightarrow \mathsf{b} \text{ iff } \bigsqcup_{i \in J} \mathsf{b}_i \sqsupseteq \mathsf{b}, \text{ where } J = \{i \in I \mid \mathsf{a} \sqsupseteq \mathsf{a}_i\} \qquad (1)$$

In view of the above given correspondence between step functions, joins and type constructors the statement (1) can be rewritten for types as follows:

$$\bigcap_{i\in I}(A_i \to B_i) \le A \to B \text{ iff } \bigcap_{i\in J} B_i \le B \text{ where } J = \{i \in I \mid A \le A_i\} \qquad (2)$$

where we use the standard notation for types and preorders on them (obtained from Definitions 10 and 14 by replacing pretypes with types). Condition (2) holds for almost all intersection type theories considered in the literature, in particular for those in the present paper and in [4], appearing also in this issue. Section 6 of [4] discusses similarities and differences between these two papers.

3 The Models

We use standard notations for λ-terms and β-reductions.

Definition 1 (The set Λ of λ-terms) *The set Λ of (type-free) λ-terms is defined by the following abstract syntax.*

$$\begin{array}{|lcl|} \Lambda & ::= & \mathtt{var} \mid (\Lambda\Lambda) \mid (\lambda\mathtt{var}\Lambda) \\ \mathtt{var} & ::= & x \mid \mathtt{var}' \end{array}$$

We use $x, y, z, \ldots, x_1, \ldots$ for arbitrary term variables and $\mathbf{t}, \mathbf{u}, \mathbf{p}, \ldots, \mathbf{t}_1, \ldots$ for arbitrary terms. When writing the terms we assume the standard conventions on vectors, parentheses and dots [7], namely $\lambda\vec{x}.\mathbf{u}\,\vec{\mathbf{t}}$ is a short for

$$(\lambda x_1 \ldots (\lambda x_n (\ldots (\mathbf{u}\mathbf{t}_1) \ldots)\mathbf{t}_k) \ldots),$$

where $\vec{\mathbf{t}}$ is $\mathbf{t}_1 \ldots \mathbf{t}_k$ and k is the *length* of $\vec{\mathbf{t}}$. The *initial abstractions* in the given term are $\lambda x_1 \ldots \lambda x_n$.

$\mathsf{FV}(\mathbf{t})$ denotes the set of free variables of a term $\mathbf{t}$. By $\mathbf{t}[x := \mathbf{u}]$ we denote the term obtained by substituting the term $\mathbf{u}$ for all free occurrences of the variable x in $\mathbf{t}$, taking into account that free occurrences of variables of $\mathbf{u}$ remain free in the term obtained.

The axiom of *β-reduction* is $(\lambda x.\mathbf{t})\mathbf{u} \to_\beta \mathbf{t}[x := \mathbf{u}]$. A term of the form $(\lambda x.\mathbf{t})\mathbf{u}$ is called a *β-redex*. The congruence induced by the transitive reflexive closure of $\to_\beta$ is denoted by $\twoheadrightarrow_\beta$: it is closed under substitutions. A term is *a normal form* if it does not contain β-redexes.

We introduce now the computational behaviours of λ-terms we want to characterise.

Definition 2 (Normalization properties)

(1) A term $\mathbf{t}$ has a normal form, $\mathbf{t} \in \mathcal{N}$*, if* $\mathbf{t}$ *reduces to a normal form.*
(2) A term $\mathbf{t}$ has a head normal form, $\mathbf{t} \in \mathcal{HN}$*, if* $\mathbf{t}$ *reduces to a term of the form* $\lambda \vec{x}.y\,\vec{\mathbf{t}}$ *(where possibly* y *appears in* $\vec{x}$*).*
(3) A term $\mathbf{t}$ has a weak head normal form, $\mathbf{t} \in \mathcal{WN}$*, if* $\mathbf{t}$ *reduces to an abstraction or to a term starting with a free variable.*

For each of the above properties, we also consider the corresponding *persistent* version (see Definition 3). *Persistently normalising* terms were introduced in Böhm and Dezani [8].

Definition 3 (Persistent normalisation properties)

(1) A term $\mathbf{t}$ *is* persistently normalising, $\mathbf{t} \in \mathcal{PN}$*, if* $\mathbf{t}\,\vec{\mathbf{u}} \in \mathcal{N}$ *for all terms* $\vec{\mathbf{u}}$ *in* $\mathcal{N}$*.*
(2) A term $\mathbf{t}$ *is* persistently head normalising, $\mathbf{t} \in \mathcal{PHN}$*, if* $\mathbf{t}\,\vec{\mathbf{u}} \in \mathcal{HN}$ *for all terms* $\vec{\mathbf{u}}$*.*
(3) A term $\mathbf{t}$ *is* persistently weak head normalising, $\mathbf{t} \in \mathcal{PWN}$*, if* $\mathbf{t}\,\vec{\mathbf{u}} \in \mathcal{WN}$ *for all terms* $\vec{\mathbf{u}}$*.*

We also consider the reducibility of terms to closed terms, to closed normal forms, and to closed head normal forms.

Definition 4 (Closability properties)

(1) A term $\mathbf{t}$ *is* closable, $\mathbf{t} \in \mathcal{C}$*, if* $\mathbf{t}$ *reduces to a closed term.*
(2) A term $\mathbf{t}$ *is* closable normalising, $\mathbf{t} \in \mathcal{CN}$*, if* $\mathbf{t}$ *reduces to a closed normal form.*
(3) A term $\mathbf{t}$ *is* closable head normalising, $\mathbf{t} \in \mathcal{CHN}$*, if* $\mathbf{t}$ *reduces to a closed head normal form.*

Example 5

Let $\mathbf{I} \equiv \lambda x.x$*,* $\mathbf{W}_2 \equiv \lambda x.xx, \mathbf{W}_4 \equiv \mathbf{W}_2\mathbf{W}_2$*,* $\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$*,* $\mathbf{K} \equiv \lambda xy.x$*.*

- $\lambda x.x\mathbf{W}_2\mathbf{W}_2 \in \mathcal{N}$*, but* $\lambda x.x\mathbf{W}_2\mathbf{W}_2 \notin \mathcal{PWN}$ *(hence* $\lambda x.x\mathbf{W}_2\mathbf{W}_2 \notin \mathcal{PHN}$*), since* $(\lambda x.x\mathbf{W}_2\mathbf{W}_2)\mathbf{I} \twoheadrightarrow_\beta \mathbf{W}_2\mathbf{W}_2 \notin \mathcal{WN}$*. Notice that* $\lambda x.x\mathbf{W}_2\mathbf{W}_2 \notin \mathcal{PN}$ *since* $\mathbf{I} \in \mathcal{N}$*. Finally,* $\lambda x.x\mathbf{W}_2\mathbf{W}_2 \in \mathcal{CN}$*.*
- $\lambda x.y\mathbf{W}_4 \in \mathcal{PHN}$*, but* $\lambda x.y\mathbf{W}_4 \notin \mathcal{N}$*.*
- $\lambda x.x\mathbf{W}_4 \in \mathcal{HN}$*, but* $\lambda x.x\mathbf{W}_4 \notin \mathcal{N}$ *and* $\lambda x.x\mathbf{W}_4 \notin \mathcal{PWN}$*, since* $(\lambda x.x\mathbf{W}_4)\mathbf{W}_2 \to_\beta \mathbf{W}_2\mathbf{W}_4 \notin \mathcal{WN}$*. Moreover* $\lambda x.x\mathbf{W}_4 \in \mathcal{CHN}$*, but* $\lambda x.x\mathbf{W}_4 \notin \mathcal{CN}$*.*
- $\mathbf{YK} \in \mathcal{PWN}$*, but* $\mathbf{YK} \notin \mathcal{HN}$*, hence* $\mathbf{YK} \notin \mathcal{PHN}$*.*
- $\lambda x.\mathbf{W}_4 \in \mathcal{WN}$*, but* $\lambda x.\mathbf{W}_4 \notin \mathcal{HN}$ *and* $\lambda x.\mathbf{W}_4 \notin \mathcal{PWN}$*, since* $(\lambda x.\mathbf{W}_4)\mathbf{t} \twoheadrightarrow_\beta \mathbf{W}_4 \notin \mathcal{WN}$*. Moreover* $\lambda x.\mathbf{W}_4 \in \mathcal{C}$*, but* $\lambda x.\mathbf{W}_4 \notin \mathcal{CHN}$*, hence* $\lambda x.\mathbf{W}_4 \notin$

$\mathcal{CN}$.

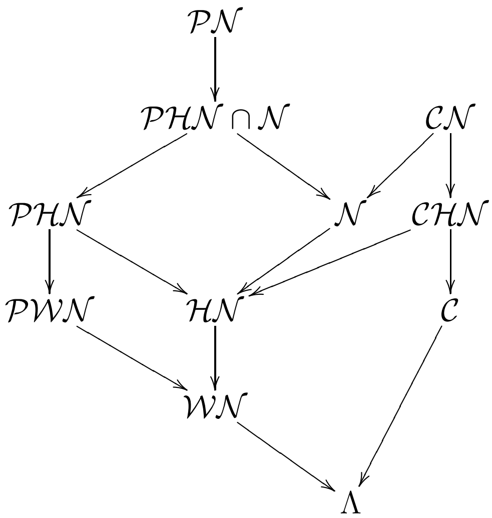

Fig. 1. Inclusions between sets of λ-terms

The following proposition, represented pictorially by Figure 1, sums up the mutual implications between the above notions.

Proposition 6 *The following* strict *inclusions hold:*

$$\begin{array}{lllll}
\mathcal{PN} & \subsetneq \mathcal{N} & \subsetneq \mathcal{HN} & \subsetneq \mathcal{WN} \subsetneq \Lambda \\
\mathcal{PN} & \subsetneq \mathcal{PHN} & \subsetneq \mathcal{PWN} & \subsetneq \mathcal{WN} \\
\mathcal{PHN} & \subsetneq \mathcal{HN} & & \\
\mathcal{CN} & \subsetneq \mathcal{CHN} & \subsetneq \mathcal{C} & \subsetneq \Lambda \\
\mathcal{CN} & \subsetneq \mathcal{N} & & \\
\mathcal{CHN} & \subsetneq \mathcal{HN}. & &
\end{array}$$

No other inclusion holds between the above sets. Moreover

$$\begin{array}{ll}
\mathcal{PHN} = \mathcal{PWN} \cap \mathcal{HN} & \mathcal{PN} \subsetneq \mathcal{PHN} \cap \mathcal{N} \\
\mathcal{CHN} = \mathcal{C} \cap \mathcal{HN} & \mathcal{CN} = \mathcal{C} \cap \mathcal{N} \\
\mathcal{C} \cap \mathcal{PHN} = \emptyset & \mathcal{C} \cap \mathcal{PN} = \emptyset.
\end{array}$$

PROOF.

A persistently weak head normalising term $\mathbf{t}$ is either an unsolvable term of order ∞ (as defined in Abramsky and Ong [2]), i.e. for all n there is $\mathbf{u}$ such that $\mathbf{t} =_\beta \lambda x_1 \dots x_n.\mathbf{u}$, or it is a solvable term such that the head variable of its head normal form is free. In fact, if $\mathbf{t}$ is an unsolvable term of finite order, i.e. $\mathbf{t} =_\beta \lambda x_1 \dots x_n.\mathbf{u}$ where $\mathbf{u}$ is unsolvable and it does not reduce to an abstraction, then $\mathbf{t}\vec{\mathbf{u}} \notin \mathcal{WN}$, where $\vec{\mathbf{u}}$ are n arbitrary λ-terms. If $\mathbf{t} =_\beta \lambda \vec{x}\, y\, \vec{z}.y\vec{\mathbf{u}}$ we get $\mathbf{t}\vec{\mathbf{x}}\mathbf{W}_4\vec{\mathbf{z}} \twoheadrightarrow_\beta \mathbf{W}_4\vec{\mathbf{u}'} \notin \mathcal{WN}$, where $\vec{\mathbf{x}}$ has the same length as $\vec{x}$, $\vec{\mathbf{z}}$ has the same length as $\vec{z}$, $\mathbf{W}_4$ is defined in Example 5, and $\vec{\mathbf{u}'} = \vec{\mathbf{u}}[\vec{x} := \vec{\mathbf{x}}, y := \mathbf{W}_4, \vec{z} := \vec{\mathbf{z}}]$.

The above discussion also shows that a persistently head normalising term is a solvable term such that the head variable of its head normal form is free. So we get:

$$\mathcal{PHN} = \mathcal{PWN} \cap \mathcal{HN}.$$

From the same example we have that a necessary condition for a normalising term to be a persistently normalising term is that the head variable of its normal form is free. This condition is not sufficient, since for example $(\lambda x.y(xx))\mathbf{W}_2 \to_\beta y\mathbf{W}_4$ ($\mathbf{W}_2$ and $\mathbf{W}_4$ are defined in Example 5). Being $\lambda x.y(xx) \in \mathcal{PHN}$ and $\lambda x.y(xx) \in \mathcal{N}$ this term shows that:

$$\mathcal{PN} \subsetneq \mathcal{PHN} \cap \mathcal{N}.$$

For closable terms we clearly have:

$$\mathcal{CHN} = \mathcal{C} \cap \mathcal{HN} \qquad \mathcal{CN} = \mathcal{C} \cap \mathcal{N}$$
$$\mathcal{C} \cap \mathcal{PHN} = \emptyset \qquad \mathcal{C} \cap \mathcal{PN} = \emptyset.$$

The above discussion gives some inclusions between the current sets of terms, and Example 5 shows differences between them. The remaining inclusions easily follow by definition.

Our goal is to build two inverse limit λ-models (Scott [28]) which satisfy the following condition:
for each one of the above nine sets of terms there is a corresponding element in one of these models such that a term belongs to the set iff its interpretation (in a suitable environment) is greater than or equal to that element.

We therefore need to discuss the functional behaviours of the terms belonging to these classes, in particular with respect to the step functions.

A weak head normalising term either reduces to an abstraction or to an application of a variable to (possibly zero) terms: in both cases (in a suitable environment) it behaves at least as well as (i.e. its interpretation is greater or equal to the interpretation of) the step function $\bot \Rightarrow \bot$. Therefore, we can choose the representative of the step function $\bot \Rightarrow \bot$ as the element which corresponds to the set $\mathcal{WN}$. We need to consider a model in which this step function is not the bottom of the whole domain, i.e. a solution of the domain equation $D = [D \to D]_\bot$, where as usual $[D \to D]$ is the domain of continuous functions from D to D and $_\bot$ is the lifting operator.

A persistently weak head normalising term applied to any number of arbitrary terms gives a weak head normalising term, i.e. it behaves at least as well as the step functions $\underbrace{\bot \Rightarrow \ldots \Rightarrow \bot}_{n} \Rightarrow \bot$ for any value n. Therefore, the element representing $\bigsqcup_{n\in\mathbb{N}}(\underbrace{\bot \Rightarrow \ldots \Rightarrow \bot}_{n} \Rightarrow \bot)$ is a good candidate for the correspondence with the set $\mathcal{PWN}$.

A head normalising term, when applied to a persistently head normalising term, reduces to a head normalising term: in its turn a persistently head normalising term applied to an arbitrary term gives a persistently head normalising term. Therefore, if h and $\hat{\mathsf{h}}$ are two elements corresponding to the sets $\mathcal{HN}$ and $\mathcal{PHN}$ respectively, they represent the step functions $\hat{\mathsf{h}} \Rightarrow \mathsf{h}$ and $\bot \Rightarrow \hat{\mathsf{h}}$.

A normalising term is also a head normalising term and therefore it behaves at least as well as the step function $\hat{\mathsf{h}} \Rightarrow \mathsf{h}$. Similarly a persistently normalising term is also a persistently head normalising term and therefore it behaves at least as well as the step function $\bot \Rightarrow \hat{\mathsf{h}}$. Moreover, a persistently normalising term applied to a normalising term gives a persistently normalising term. One can show that:

Proposition 7 *The application of a normalising term to a persistently normalising term is in turn a normalising term.*

PROOF.

We show that if $\mathbf{u} \in \mathcal{N}$ and $\mathbf{t} \in \mathcal{PN}$, then $\mathbf{ut} \in \mathcal{N}$. We can assume that $\mathbf{u}$ is in normal form. If $\mathbf{u}$ is λ-free it is trivial. Otherwise let $\mathbf{u} \equiv \lambda x.\mathbf{u}'$. The proof is by induction on the number of occurrences of x in $\mathbf{u}'$. The basic step, i.e. x does not occur in $\mathbf{u}'$, is immediate. If x occurs in $\mathbf{u}'$, let $\mathbf{u}' \equiv C[x]$, where the hole in $C[\;]$ identifies the left-most occurrence of x in $\mathbf{u}'$. Let y be fresh: by the induction hypothesis $(\lambda x.C[y])\mathbf{t} \twoheadrightarrow_\beta C'[y]$ and $C'[y]$ is in normal form. By construction there is exactly one hole in $C'[\;]$. Let $\vec{\mathbf{u}}$ be all the terms to which $[\;]$ is applied in $C'[\;]$. Since $\mathbf{t} \in \mathcal{PN}$, $\mathbf{t}\vec{\mathbf{u}} \in \mathcal{N}$ and therefore $(\lambda y.C'[y])\mathbf{t} \in \mathcal{N}$ too. We conclude $\mathbf{ut} \in \mathcal{N}$ since $\mathbf{ut} =_\beta (\lambda xy.C[y])\mathbf{tt} =_\beta (\lambda y.C'[y])\mathbf{t}$.

Therefore, if n and $\hat{\mathsf{n}}$ are two elements corresponding respectively to the sets $\mathcal{N}$ and $\mathcal{PN}$, they represent the functions $(\hat{\mathsf{h}} \Rightarrow \mathsf{h}) \sqcup (\hat{\mathsf{n}} \Rightarrow \mathsf{n})$ and $(\bot \Rightarrow \hat{\mathsf{h}}) \sqcup (\mathsf{n} \Rightarrow \hat{\mathsf{n}})$.

A closable term applied to a closable term reduces to a closable term. Then, if c is the element representing the set $\mathcal{C}$, it behaves like the function $\mathsf{c} \Rightarrow \mathsf{c}$. The key observation here is that there are closable terms (like $\mathbf{W}_4$, see Example 5) which are not weak head normalising, and therefore we need to equate $\bot$ and $\bot \Rightarrow \bot$, i.e. we need to consider a solution of the domain equation $D = [D \rightarrow D]$. Moreover we do not have a join between c and $\hat{\mathsf{h}}$ (and hence $\hat{\mathsf{n}}$) since all persistently head normalising terms are open.

To sum up we consider a lattice $\mathcal{D}_0$ with elements $\hat{\mathsf{n}}, \hat{\mathsf{h}} \sqcup \mathsf{n}, \hat{\mathsf{h}}, \mathsf{n}, \mathsf{h}, \bot$ and a cpo $\mathcal{E}_0$ obtained by adding to $\mathcal{D}_0$ the element c and the relative joins (see Figure 2).

We suggest the reader to compare the inclusions between sets of λ-terms (Figure 1) with the cpo $\mathcal{E}_0$ (Figure 2): this makes apparent the correspondence between properties of terms and elements of $\mathcal{E}_0$.

In Section 2 we recalled that the construction of inverse limit λ-models is parametric in the initial cpos and the embedding between them. So in the following definition we take particular $\mathcal{D}_0, \mathcal{D}_1$ and $\mathcal{E}_0, \mathcal{E}_1$, as well as $\mathsf{i}_0^{\mathcal{D}}$ and $\mathsf{i}_0^{\mathcal{E}}$ in order to build two inverse limit λ-models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$.

Definition 8

(1) Let $\mathcal{D}_\infty$ be the inverse limit λ-model obtained by taking as $\mathcal{D}_0$ the lattice in Figure 2, as $\mathcal{D}_1$ the lattice $[\mathcal{D}_0 \to \mathcal{D}_0]_\bot$, and by defining the embedding $\mathsf{i}_0^{\mathcal{D}} : \mathcal{D}_0 \to [\mathcal{D}_0 \to \mathcal{D}_0]_\bot$ as follows:

$$\mathsf{i}_0^{\mathcal{D}}(\hat{\mathsf{n}}) = (\bot \Rightarrow \hat{\mathsf{h}}) \sqcup (\mathsf{n} \Rightarrow \hat{\mathsf{n}}), \quad \mathsf{i}_0^{\mathcal{D}}(\mathsf{n}) = (\hat{\mathsf{h}} \Rightarrow \mathsf{h}) \sqcup (\hat{\mathsf{n}} \Rightarrow \mathsf{n}),$$
$$\mathsf{i}_0^{\mathcal{D}}(\hat{\mathsf{h}}) = \bot \Rightarrow \hat{\mathsf{h}}, \qquad \mathsf{i}_0^{\mathcal{D}}(\mathsf{h}) = \hat{\mathsf{h}} \Rightarrow \mathsf{h}, \qquad \mathsf{i}_0^{\mathcal{D}}(\bot) = \bot.$$

(2) Let $\mathcal{E}_\infty$ be the inverse limit λ-model obtained by taking as $\mathcal{E}_0$ the cpo in Figure 2, as $\mathcal{E}_1$ the cpo $[\mathcal{E}_0 \to \mathcal{E}_0]$, and by defining the embedding $\mathsf{i}_0^{\mathcal{E}} : \mathcal{E}_0 \to [\mathcal{E}_0 \to \mathcal{E}_0]$ as follows:

$$\mathsf{i}_0^{\mathcal{E}}(\hat{\mathsf{n}}) = (\bot \Rightarrow \hat{\mathsf{h}}) \sqcup (\mathsf{n} \Rightarrow \hat{\mathsf{n}}), \quad \mathsf{i}_0^{\mathcal{E}}(\mathsf{n}) = (\hat{\mathsf{h}} \Rightarrow \mathsf{h}) \sqcup (\hat{\mathsf{n}} \Rightarrow \mathsf{n}),$$
$$\mathsf{i}_0^{\mathcal{E}}(\hat{\mathsf{h}}) = \bot \Rightarrow \hat{\mathsf{h}}, \qquad \mathsf{i}_0^{\mathcal{E}}(\mathsf{h}) = \hat{\mathsf{h}} \Rightarrow \mathsf{h},$$
$$\mathsf{i}_0^{\mathcal{E}}(\mathsf{c}) = \mathsf{c} \Rightarrow \mathsf{c}, \qquad \mathsf{i}_0^{\mathcal{E}}(\bot) = \bot \Rightarrow \bot.$$

(3) We will denote the partial orders on $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ by $\sqsubseteq^{\mathcal{D}}$ and $\sqsubseteq^{\mathcal{E}}$, respectively.

Since each variable is clearly a persistently normalising term, it is meaningful to interpret terms in the environment which maps each variable to the element $\hat{\mathsf{n}}$. The main result of our paper is:

Theorem 9 (Main Theorem, Version I) *Let $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ be the inverse limit λ-models defined in Definition 8 and $\rho_{\hat{\mathsf{n}}}$ the environment defined by $\rho_{\hat{\mathsf{n}}}(x) = \hat{\mathsf{n}}$ for all $x \in \mathtt{var}$. Then:*

(1) $\mathsf{t} \in \mathcal{PN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \hat{\mathsf{n}}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \hat{\mathsf{n}}$;
(2) $\mathsf{t} \in \mathcal{N}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \mathsf{n}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{n}$;
(3) $\mathsf{t} \in \mathcal{PHN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \hat{\mathsf{h}}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \hat{\mathsf{h}}$;
(4) $\mathsf{t} \in \mathcal{HN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \mathsf{h}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{h}$;
(5) $\mathsf{t} \in \mathcal{PWN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \bigsqcup_{n \in \mathbb{N}} (\underbrace{\bot \Rightarrow \ldots \Rightarrow \bot}_{n} \Rightarrow \bot)$;

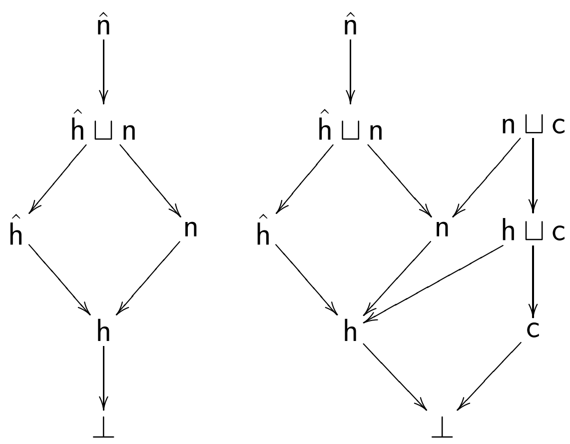

Fig. 2. The lattice $\mathcal{D}_0$ and the cpo $\mathcal{E}_0$

(6) $\mathsf{t} \in \mathcal{WN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{D}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \bot \Rightarrow \bot$*;*
(7) $\mathsf{t} \in \mathcal{CN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c} \sqcup \mathsf{n}$*;*
(8) $\mathsf{t} \in \mathcal{CHN}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c} \sqcup \mathsf{h}$*;*
(9) $\mathsf{t} \in \mathcal{C}$ *iff* $[\![\mathsf{t}]\!]^{\mathcal{E}_\infty}_{\rho_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c}$.

The proof of this theorem is done by means of finitary logical descriptions of $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ obtained by defining intersection type assignment systems in Section 5.

4 Types and Filter Models

In the present section we will exploit the duality described in Section 2 for building finitary logic descriptions of the two inverse limit λ-models introduced in Definition 8.

Let $\widetilde{\mathcal{E}}_0$ be the lattice obtained from $\mathcal{E}_0$ by adding the missing joins and $\widetilde{\mathcal{E}}_\infty$ the inverse limit λ-model obtained from $\widetilde{\mathcal{E}}_0$ by taking as $\widetilde{\mathcal{E}}_1$ the cpo $[\widetilde{\mathcal{E}}_0 \to \widetilde{\mathcal{E}}_0]$, and as initial embedding the embedding $\mathsf{i}^{\mathcal{E}}_0$ of Definition 8. We first define pretypes corresponding to the elements of $\widetilde{\mathcal{E}}_\infty$ and then types corresponding to the elements of $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$.

Definition 10 (The set $\mathbb{P}$ of pretypes) *The set $\mathbb{P}$ of pretypes is defined as follows.*

$$\mathbb{P} \quad ::= \quad \nu \mid \hat{\nu} \mid \mu \mid \hat{\mu} \mid \gamma \mid \Omega \mid \mathbb{P} \to \mathbb{P} \mid \mathbb{P} \cap \mathbb{P}$$

Pretypes will be denoted by $\mathrm{A}, \mathrm{B}, \mathrm{A}_1, \ldots, \mathrm{A}', \ldots$.

We give now the correspondence between pretypes and the finite elements of $\widetilde{\mathcal{E}}_\infty$ (as usual we identify an element of $\widetilde{\mathcal{E}}_n$ with its embedding in $\widetilde{\mathcal{E}}_\infty$).

(*refl*)	$\mathrm{A} \le \mathrm{A}$	(*trans*)	$\dfrac{\mathrm{A} \le \mathrm{B} \quad \mathrm{B} \le \mathrm{C}}{\mathrm{A} \le \mathrm{C}}$
(*mon*)	$\dfrac{\mathrm{A} \le \mathrm{A}' \quad \mathrm{B} \le \mathrm{B}'}{\mathrm{A} \cap \mathrm{B} \le \mathrm{A}' \cap \mathrm{B}'}$	(*idem*)	$\mathrm{A} \le \mathrm{A} \cap \mathrm{A}$
(*incl*$_L$)	$\mathrm{A} \cap \mathrm{B} \le \mathrm{A}$	(*incl*$_R$)	$\mathrm{A} \cap \mathrm{B} \le \mathrm{B}$
($\to\cap$)	$(\mathrm{A} \to \mathrm{B}) \cap (\mathrm{A} \to \mathrm{C}) \le \mathrm{A} \to \mathrm{B} \cap \mathrm{C}$	(η)	$\dfrac{\mathrm{A}' \le \mathrm{A} \quad \mathrm{B} \le \mathrm{B}'}{\mathrm{A} \to \mathrm{B} \le \mathrm{A}' \to \mathrm{B}'}$
(Ω)	$\mathrm{A} \le \Omega$		
($\hat{\nu}\nu$)	$\hat{\nu} \le \nu$	($\hat{\nu}\hat{\mu}$)	$\hat{\nu} \le \hat{\mu}$
($\nu\mu$)	$\nu \le \mu$	($\hat{\mu}\mu$)	$\hat{\mu} \le \mu$
($\hat{\nu}\to$)	$\hat{\nu} \sim (\Omega \to \hat{\mu}) \cap (\nu \to \hat{\nu})$	($\nu\to$)	$\nu \sim (\hat{\mu} \to \mu) \cap (\hat{\nu} \to \nu)$
($\hat{\mu}\to$)	$\hat{\mu} \sim \Omega \to \hat{\mu}$	($\mu\to$)	$\mu \sim \hat{\mu} \to \mu$

where $\mathrm{A} \sim \mathrm{B}$ is short for $\mathrm{A} \le \mathrm{B}$ and $\mathrm{B} \le \mathrm{A}$.

Fig. 3. Preorder axioms and rules

Definition 11 (The mapping $\mathtt{m}$) *The mapping* $\mathtt{m} : \mathbb{P} \to \widetilde{\mathcal{E}}_\infty$ *is defined as follows.*

$$\begin{array}{ll} \mathtt{m}(\nu) = \mathsf{n} & \mathtt{m}(\hat{\nu}) = \hat{\mathsf{n}} \\ \mathtt{m}(\mu) = \mathsf{h} & \mathtt{m}(\hat{\mu}) = \hat{\mathsf{h}} \\ \mathtt{m}(\gamma) = \mathsf{c} & \mathtt{m}(\Omega) = \bot \\ \mathtt{m}(\mathrm{A} \to \mathrm{B}) = \mathtt{m}(\mathrm{A}) \Rightarrow \mathtt{m}(\mathrm{B}) & \mathtt{m}(\mathrm{A} \cap \mathrm{B}) = \mathtt{m}(\mathrm{A}) \sqcup \mathtt{m}(\mathrm{B}). \end{array}$$

The mapping $\mathtt{m}$ allows us to single out the sets of types.

Definition 12 (The sets $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$ of types)

(1) A pretype A *is a $\mathcal{D}$-type,* $\mathrm{A} \in \mathbb{T}^{\mathcal{D}}$ *iff* $\mathtt{m}(\mathrm{A}) \in \mathcal{D}_\infty$*;*
(2) A pretype A *is an $\mathcal{E}$-type,* $\mathrm{A} \in \mathbb{T}^{\mathcal{E}}$ *iff* $\mathtt{m}(\mathrm{A}) \in \mathcal{E}_\infty$*.*

Types will be denoted by $A, B, \ldots, A_1, \ldots$. When writing types we shall use the following convention: the constructor $\cap$ takes precedence over the constructor $\to$ which associates to the right. For example

$$(A \to B \to C) \cap A \to B \to C \equiv ((A \to (B \to C)) \cap A) \to (B \to C).$$

Moreover $A^n \to B$ will be short for $\underbrace{A \to \cdots \to A}_{n} \to B$ $(n \ge 0)$.

Remark 13 *Definition 12 makes intersection on types a partial operation. Actually, given types A, B, the pretype $A \cap B$ is a type iff the compact elements $\mathtt{m}(A), \mathtt{m}(B)$ have a join.*

We can now give the preorders on pretypes and on types.

Definition 14 (Preorder on pretypes) *The relation $\leq$ on pretypes is defined by the axioms and rules given in Figure 3.*

In Figure 3 the first eight axioms and rules correspond to standard properties of joins and step functions. The successive five axioms mimic the partial order on $\mathcal{D}_0$. The last four axioms encode the initial embedding of the constants different from γ and Ω.

Being $\cap$ commutative and associative, we will write $\bigcap_{i\leq n} A_i$ for $A_1 \cap \ldots \cap A_n$. Similarly we will write $\bigcap_{i\in I} A_i$. We convene that I, J, K etc., when referred to as sets of indices for types, always denote finite sets and that $\bigcap_{i\in\emptyset} A_i$ is Ω.

Definition 15 (Preorders on $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$) *(1) The relation $\leq_{\mathcal{D}}$ is defined on $\mathbb{T}^{\mathcal{D}}$ by the axioms and rules of Figure 3 plus the following axiom:*

$$(\Omega\!\to) \qquad A \to \Omega \leq \Omega \to \Omega;$$

(2) The relation $\leq_{\mathcal{E}}$ is defined on $\mathbb{T}^{\mathcal{E}}$ by the axioms and rules of Figure 3 plus the following axioms:

$$(\Omega\text{-}\eta) \quad \Omega \leq \Omega \to \Omega \qquad (\gamma\!\to) \quad \gamma \sim \gamma \to \gamma.$$

The axioms $(\Omega\!\to)$ and $(\Omega\text{-}\eta)$ reflect the differences between the embeddings $\mathsf{i}_0^{\mathcal{D}}$ and $\mathsf{i}_0^{\mathcal{E}}$ on $\bot$. Notice that $(\Omega\text{-}\eta)$ and (Ω) imply $(\Omega\!\to)$. The axiom $(\gamma\!\to)$ encodes the initial embedding of the constant γ.

Remark 16 *The partial orders induced by the preorders $\leq_{\mathcal{D}}$ and $\leq_{\mathcal{E}}$ do not collapse the sets of types $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$ to a single point: this is a consequence of Lemma 44 in Section 7.*
The sets $\mathbb{T}^{\mathcal{D}}$ and $\mathbb{T}^{\mathcal{E}}$ are not downward closed under $\leq$, e.g $\hat{\nu} \in \mathbb{T}^{\mathcal{D}}$, $\hat{\nu} \in \mathbb{T}^{\mathcal{E}}$ and by rule $(incl_L)$ $\hat{\nu} \cap \gamma \leq \hat{\nu}$, but $\hat{\nu} \cap \gamma \notin \mathbb{T}^{\mathcal{D}}$, $\hat{\nu} \cap \gamma \notin \mathbb{T}^{\mathcal{E}}$.

We build filters on the set of pretypes and then single out the filters on the sets of types.

Definition 17 (The sets $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{F}^{\mathcal{E}}$) *(1) A* filter *is a set $\Xi \subseteq \mathbb{P}$ such that:*

(a) $\Omega \in \Xi$;
(b) if $A \leq B$ and $A \in \Xi$, then $B \in \Xi$;
(c) if $A, B \in \Xi$, then $A \cap B \in \Xi$;

where $\leq$ is the preorder defined in Figure 3;

(2) if $\Xi \subseteq \mathbb{P}$, then $\uparrow \Xi$ denotes the filter generated by Ξ;

(3) a filter is principal *if it is of the shape $\uparrow \{A\}$, for some type A. We shall denote $\uparrow \{A\}$ simply by $\uparrow A$;*

(4) $\mathcal{F}^{\mathcal{D}}$ denotes the set of filters Ξ such that

(a) if $A \in \Xi$, then $A \in \mathbb{T}^{\mathcal{D}}$;
(b) if $A \leq_{\mathcal{D}} B$ and $A \in \Xi$, then $B \in \Xi$;

(5) $\mathcal{F}^{\mathcal{E}}$ denotes the set of filters Ξ such that
(a) if $A \in \Xi$, then $A \in \mathbb{T}^{\mathcal{E}}$;
(b) if $A \leq_{\mathcal{E}} B$ and $A \in \Xi$, then $B \in \Xi$.

It is easy to verify that both $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{F}^{\mathcal{E}}$, ordered by subset inclusion, are Scott domains. As is well known, the compact elements are precisely the principal filters, and the bottom element is $\uparrow \Omega$. Further, $\mathcal{F}^{\mathcal{D}}$ is an ω-algebraic complete lattice, since it has the top element $\mathbb{T}^{\mathcal{D}}$.

Using the mapping $\mathtt{m}$ we can show that $\mathcal{F}^{\mathcal{D}}$ and $\mathcal{D}_\infty$ are isomorphic as ω-algebraic complete lattices, and that $\mathcal{F}^{\mathcal{E}}$ and $\mathcal{E}_\infty$ are isomorphic as Scott domains. In this respect, it is useful to show that the mapping $\mathtt{m}$ agrees with the preorders on types and the partial orders on inverse limit λ-models. Let $\nabla \in \{\mathcal{D}, \mathcal{E}\}$.

Lemma 18 *For all types $A, B \in \mathbb{T}^{\nabla}$ we have:*

$$\mathtt{m}(A) \sqsupseteq^{\nabla} \mathtt{m}(B) \text{ iff } A \leq_{\nabla} B.$$

PROOF.

For the reader's convenience we rewrite here the statement (1) of Section 2:

$$\bigsqcup_{i\in I}(\mathsf{a}_i \Rightarrow \mathsf{b}_i) \sqsupseteq \mathsf{a} \Rightarrow \mathsf{b} \text{ iff } \bigsqcup_{i\in J} \mathsf{b}_i \sqsupseteq \mathsf{b}, \text{ where } J = \{i \in I \mid \mathsf{a} \sqsupseteq \mathsf{a}_i\} \qquad (3)$$

Notice the set J can be empty and in this case $\mathsf{b} = \perp$.

We show that $A \leq_{\nabla} B$ implies $\mathtt{m}(A) \sqsupseteq^{\nabla} \mathtt{m}(B)$ by induction on $\leq_{\nabla}$. The axioms on constant types just mimic the order on the initial cpos and the initial embeddings. Axiom $(\Omega\rightarrow)$ immediately follows from the definition of the step function. The only interesting cases are axiom $(\rightarrow\cap)$ and rule (η).

For axiom $(\rightarrow\cap)$ we have to show that

$$\mathtt{m}(A \rightarrow B \cap C) \sqsubseteq^{\nabla} \mathtt{m}((A \rightarrow B) \cap (A \rightarrow C))$$

which is equivalent to

$$\mathtt{m}(A) \Rightarrow (\mathtt{m}(B) \sqcup \mathtt{m}(C)) \sqsubseteq^{\nabla} (\mathtt{m}(A) \Rightarrow \mathtt{m}(B)) \sqcup (\mathtt{m}(A) \Rightarrow \mathtt{m}(C)).$$

Then, the statement follows by (3), taking $\mathsf{a}_1 = \mathsf{a}_2 = \mathsf{a} = \mathtt{m}(A), \mathsf{b}_1 = \mathtt{m}(B), \mathsf{b}_2 = \mathtt{m}(C), \mathsf{b} = \mathtt{m}(B) \sqcup \mathtt{m}(C)$.

For rule (η), let $A' \leq_\nabla A$ and $B \leq_\nabla B'$. By the induction hypothesis $\mathtt{m}(A) \sqsubseteq^\nabla \mathtt{m}(A')$ and $\mathtt{m}(B') \sqsubseteq^\nabla \mathtt{m}(B)$, hence by (3) $\mathtt{m}(A') \Rightarrow \mathtt{m}(B') \sqsubseteq^\nabla \mathtt{m}(A) \Rightarrow \mathtt{m}(B)$. Thus we get $\mathtt{m}(A' \to B') \sqsubseteq^\nabla \mathtt{m}(A \to B)$ by the definition of $\mathtt{m}$.

Now we show that $\mathtt{m}(A) \sqsupseteq^\nabla \mathtt{m}(B)$ implies $A \leq_\nabla B$ by structural induction on A and B. First notice that each type is an intersection of type constants and arrows. Moreover, each type constant (different from Ω in case of $\leq_\mathcal{D}$) is equivalent to an intersection of arrows between constants. So each type in $\mathbb{T}^\mathcal{D}$ is either an intersection of Ω or it is equivalent to an intersection of arrows, each type in $\mathbb{T}^\mathcal{E}$ is equivalent to an intersection of arrows. More precisely, if A is not an intersection of Ω or $\nabla = \mathcal{E}$ then $A \sim^\nabla \bigcap_{i \in I}(B_i \to C_i)$ for some I, B_i, C_i such that each B_i, C_i is either a constant or a subtype of A.

If both A and B are type constants then both $\mathtt{m}(A)$ and $\mathtt{m}(B)$ are elements of the initial cpos: just notice that there is a one-one correspondence between the preorder on type constants and the partial order on the initial cpos.

If B is an intersection of Ω the proof is immediate. If A is an intersection of Ω then $\mathtt{m}(A) = \bot$ and this implies $\mathtt{m}(B) = \bot$. If $\nabla = \mathcal{D}$ it is easy to verify that only intersections of Ω are mapped into $\bot$. So B must also be an intersection of Ω.

Otherwise we get $A \sim^\nabla \bigcap_{i \in I}(A_i^{(1)} \to A_i^{(2)})$, $B \sim^\nabla \bigcap_{l \in L}(B_l^{(1)} \to B_l^{(2)})$, for some $I, A_i^{(1)}, A_i^{(2)}, L, B_l^{(1)}, B_l^{(2)}$ such that each $A_i^{(1)}, A_i^{(2)}$ is either a constant or a subtype of A and each $B_l^{(1)}, B_l^{(2)}$ is either a constant or a subtype of B. Let $\mathtt{m}(A_i^{(h)}) = \mathsf{a}_i^{(h)}$, $\mathtt{m}(B_l^{(h)}) = \mathsf{b}_l^{(h)}$, $h = 1, 2$. Then, since by above equivalent types are mapped into the same element, $\mathtt{m}(A) = \bigsqcup_{i \in I}(\mathsf{a}_i^{(1)} \Rightarrow \mathsf{a}_i^{(2)})$ and $\mathtt{m}(B) = \bigsqcup_{l \in L}(\mathsf{b}_l^{(1)} \Rightarrow \mathsf{b}_l^{(2)})$. By (3), $\mathtt{m}(A) \sqsupseteq^\nabla \mathtt{m}(B)$ implies that for each $l \in L$ $\bigsqcup_{i \in J_l} \mathsf{a}_i^{(2)} \sqsupseteq^\nabla \mathsf{b}_l^{(2)}$ where $J_l = \{i \in I \mid \mathsf{b}_l^{(1)} \sqsupseteq^\nabla \mathsf{a}_i^{(1)}\}$. By structural induction on types, we get that $\bigcap_{i \in J_l} A_i^{(2)} \leq_\nabla B_l^{(2)}$ for each $l \in L$, where $J_l = \{i \in I \mid B_l^{(1)} \leq_\nabla A_i^{(1)}\}$. By ($\eta$) rule $\bigcap_{i \in J_l} A_i^{(1)} \to \bigcap_{i \in J_l} A_i^{(2)} \leq_\nabla B_l^{(1)} \to B_l^{(2)}$. It is easily provable that $(A_p^{(1)} \to A_p^{(2)}) \cap (A_q^{(1)} \to A_q^{(2)}) \leq_\nabla (A_p^{(1)} \cap A_q^{(1)}) \to (A_p^{(2)} \cap A_q^{(2)})$ for any $p, q \in \mathbb{N}$. Therefore, $\bigcap_{i \in I}(A_i^{(1)} \to A_i^{(2)}) \leq_\nabla \bigcap_{i \in J_l} A_i^{(1)} \to \bigcap_{i \in J_l} A_i^{(2)}$, and we can conclude $A \leq_\nabla B$.

Theorem 19 (Isomorphism)

(1) The mapping $\mathtt{m}^* : \mathcal{F}^\mathcal{D} \mapsto \mathcal{D}_\infty$ *defined by*

$$\mathtt{m}^*(\Xi) = \bigsqcup_{A \in \Xi} \mathtt{m}(A)$$

is an isomorphism between $\mathcal{F}^\mathcal{D}$ *and* $\mathcal{D}_\infty$*.*

(2) The mapping $\mathtt{m}^* : \mathcal{F}^\mathcal{E} \mapsto \mathcal{E}_\infty$ *defined by*

$$\mathtt{m}^*(\Xi) = \bigsqcup_{A \in \Xi} \mathtt{m}(A)$$

is an isomorphism between $\mathcal{F}^{\mathcal{E}}$ and $\mathcal{E}_\infty$.

PROOF. Notice that $\mathtt{m}^*(\uparrow A) = \mathtt{m}(A)$. Then the result is obvious since the mapping is an order-preserving bijection between the compact elements.

5 The Type Assignment Systems

Due to the above isomorphism the interpretations of λ-terms in $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are isomorphic to the filters of types one can derive in the following type assignment systems. This gives us the finitary logical descriptions of the models.

First we introduce bases and some related notations. Let $\nabla \in \{\mathcal{D}, \mathcal{E}\}$.

Definition 20 (Basis) *A* ∇-basis *is a (possibly infinite) set of statements of the shape $x : A$, where $A \in \mathbb{T}^\nabla$, with all term variables x distinct.*

We will use the following notations:

- If Γ is a ∇-basis then $x \in \Gamma$ is short for $x : A \in \Gamma$ for some A.
- If Γ is a ∇-basis and $A \in \mathbb{T}^\nabla$ then $\Gamma, x : A$ is short for $\Gamma \cup \{x : A\}$ when $x \notin \Gamma$.

In the following, we will use bases which assign to all variables the same type as well as bases which assign to all variables but one the same type.

Definition 21
$\Gamma_A = \{x : A \mid x \in \mathtt{var}\}$ *and* $\Gamma_A^{x:B} = \{x : B\} \cup \{y : A \mid \; y \in \mathtt{var}$ *and* $y \not\equiv x\}$.

Definition 22 (The type assignment systems) *The ∇-type assignment system is a formal system for deriving judgements of the form $\Gamma \vdash^\nabla \mathbf{t} : B$, where the* subject $\mathbf{t}$ *is an untyped λ-term, the* predicate B *is in $\mathbb{T}^\nabla$, and Γ is a ∇-basis. The system has the following axioms and rules.*

$$(\mathrm{Ax})\ \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad (\mathrm{Ax}\text{-}\Omega)\ \Gamma \vdash \mathbf{t} : \Omega$$

$$(\to \mathrm{I})\ \frac{\Gamma, x : A \vdash \mathbf{t} : B}{\Gamma \vdash \lambda x.\mathbf{t} : A \to B} \qquad (\to \mathrm{E})\ \frac{\Gamma \vdash \mathbf{t} : A \to B \quad \Gamma \vdash \mathbf{u} : A}{\Gamma \vdash \mathbf{tu} : B}$$

$$(\cap \mathrm{I})\ \frac{\Gamma \vdash \mathbf{t} : A \quad \Gamma \vdash \mathbf{t} : B}{\Gamma \vdash \mathbf{t} : A \cap B} \qquad (\leq_\nabla)\ \frac{\Gamma \vdash \mathbf{t} : A \quad A \leq_\nabla B}{\Gamma \vdash \mathbf{t} : B}$$

This way we obtain two type assignment systems in which the derivability is denoted by $\vdash^{\mathcal{D}}$ and $\vdash^{\mathcal{E}}$.

Example 23 *Figure 4 gives some paradigmatic examples of deductions in our type systems. Notice the use of intersection introduction and subsumption rules in order to derive atomic types. All derivations but the last one are valid in both systems, whereas the last one is valid only in* $\vdash^{\mathcal{E}}$. *We omit the indexes* $\mathcal{D}$ *and* $\mathcal{E}$.

Having axiom (Ax-Ω), one can give a type to a term without assigning types to its free variables. In this axiom, term M can be any λ-term. For example, in derivations $(\mathsf{D3})$ and $(\mathsf{D4})$ (Figure 4) this axiom is used to type the term $\mathbf{W}_4$ with the type Ω.

Remark 24 *Notice that we do not need restrictions in rule* $(\cap\mathrm{I})$ *also for the system* $\vdash^{\mathcal{E}}$, *since we have* $A \cap B \in \mathbb{T}^{\mathcal{E}}$ *whenever* $\Gamma \vdash^{\mathcal{E}} \mathbf{t} : A$ *and* $\Gamma \vdash^{\mathcal{E}} \mathbf{t} : B$. *This can be proved by induction on* $\mathbf{t}$ *using the Generation Theorem (Theorem 25). A shorter semantic proof is the following:* $\Gamma \vdash^{\mathcal{E}} \mathbf{t} : A$ *and* $\Gamma \vdash^{\mathcal{E}} \mathbf{t} : B$ *imply by Theorem 26* $[\![\mathbf{t}]\!]^{\mathcal{E}_\infty}_\rho \sqsupseteq^{\mathcal{E}} \mathtt{m}(A)$ *and* $[\![\mathbf{t}]\!]^{\mathcal{E}_\infty}_\rho \sqsupseteq^{\mathcal{E}} \mathtt{m}(B)$ *where* $\rho(x) = \uparrow^{\mathcal{E}} C$ *for* $(x : C) \in \Gamma$. *Therefore,* $\mathtt{m}(A)$ *and* $\mathtt{m}(B)$ *have a join,* $\mathcal{E}_\infty$ *being a cpo, i.e.* $\mathtt{m}(A) \sqcup \mathtt{m}(B) = \mathtt{m}(A \cap B)$, *and so by Definition 12* $A \cap B \in \mathbb{T}^{\mathcal{E}}$.

As usual we consider λ-terms modulo α-conversion. It is easy to verify that the intersection elimination rules are derivable [3]:

$$(\cap\mathrm{E})\ \frac{\Gamma \vdash \mathbf{t} : A \cap B}{\Gamma \vdash \mathbf{t} : A} \qquad \frac{\Gamma \vdash \mathbf{t} : A \cap B}{\Gamma \vdash \mathbf{t} : B}$$

and that the following rules are admissible:

$$(\leq_\nabla \mathrm{L})\ \frac{\Gamma, x : A \vdash \mathbf{t} : B \quad A' \leq_\nabla A}{\Gamma, x : A' \vdash \mathbf{t} : B}$$

$$(\mathrm{W})\ \frac{\Gamma \vdash \mathbf{t} : B \quad x \notin \Gamma}{\Gamma, x : A \vdash \mathbf{t} : B} \qquad (\mathrm{S})\ \frac{\Gamma, x : A \vdash \mathbf{t} : B \quad x \notin FV(\mathbf{t})}{\Gamma \vdash \mathbf{t} : B}$$

As usual we have a Generation Theorem for our type assignment systems: the proof by induction on derivations follows the proof of the same property for the standard intersection type system (see e.g. [4]).

Theorem 25 (Generation Theorem)

(1) Assume $A \not\sim^\nabla \Omega$. *Then* $\Gamma \vdash^\nabla x : A$ *iff* $(x : B) \in \Gamma$ *and* $B \leq_\nabla A$ *for some* $B \in \mathbb{T}^\nabla$.
(2) $\Gamma \vdash^\nabla \mathbf{tu} : A$ *iff* $\Gamma \vdash^\nabla \mathbf{t} : B \to A$, *and* $\Gamma \vdash^\nabla \mathbf{u} : B$ *for some* $B \in \mathbb{T}^\nabla$.
(3) $\Gamma \vdash^\nabla \lambda x.\mathbf{t} : A$ *iff* $\Gamma, x : B_i \vdash^\nabla \mathbf{t} : C_i$ *and* $\bigcap_{i \in I}(B_i \to C_i) \leq_\nabla A$, *for some* I *and* $B_i, C_i \in \mathbb{T}^\nabla$.

[3] A rule is *derivable* in a system if, for each instance of the rule, there is a deduction in the system of its conclusion from its premises. 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.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\Omega}_{\hat\nu}\vdash y:\hat\nu}\,(\mathrm{Ax})}{\Gamma^{x:\Omega}_{\hat\nu}\vdash y:\Omega\to\hat\mu}\,(\leq) \quad \dfrac{}{\Gamma^{x:\Omega}_{\hat\nu}\vdash x:\Omega}\,(\mathrm{Ax})}{\Gamma^{x:\Omega}_{\hat\nu}\vdash yx:\hat\mu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.yx:\Omega\to\hat\mu}\,(\to\mathrm{I}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\nu}_{\hat\nu}\vdash y:\hat\nu}\,(\mathrm{Ax})}{\Gamma^{x:\nu}_{\hat\nu}\vdash y:\nu\to\hat\nu}\,(\leq) \quad \dfrac{}{\Gamma^{x:\nu}_{\hat\nu}\vdash x:\nu}\,(\mathrm{Ax})}{\Gamma^{x:\nu}_{\hat\nu}\vdash yx:\hat\nu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.yx:\nu\to\hat\nu}\,(\to\mathrm{I})}{\dfrac{\Gamma_{\hat\nu}\vdash\lambda x.yx:(\Omega\to\hat\mu)\cap(\nu\to\hat\nu)}{(\mathsf{D1})\quad \Gamma_{\hat\nu}\vdash\lambda x.yx:\hat\nu}\,(\leq)}\,(\cap\mathrm{I})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x:\hat\mu}\,(\mathrm{Ax})}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x:\hat\mu\to\mu}\,(\leq) \quad \dfrac{}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x:\hat\mu}\,(\mathrm{Ax})}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash xx:\mu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.xx:\hat\mu\to\mu}\,(\to\mathrm{I}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat\nu}\vdash x:\hat\nu}\,(\mathrm{Ax})}{\Gamma_{\hat\nu}\vdash x:\hat\nu\to\nu}\,(\leq) \quad \dfrac{}{\Gamma_{\hat\nu}\vdash x:\hat\nu}\,(\mathrm{Ax})}{\Gamma_{\hat\nu}\vdash xx:\nu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.xx:\hat\nu\to\nu}\,(\to\mathrm{I})}{\dfrac{\Gamma_{\hat\nu}\vdash\lambda x.xx:(\hat\mu\to\mu)\cap(\hat\nu\to\nu)}{(\mathsf{D2})\quad \Gamma_{\hat\nu}\vdash\lambda x.xx:\nu}\,(\leq)}\,(\cap\mathrm{I})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\Omega}_{\hat\nu}\vdash y:\hat\nu}\,(\mathrm{Ax})}{\Gamma^{x:\Omega}_{\hat\nu}\vdash y:\Omega\to\hat\mu}\,(\leq) \quad \dfrac{}{\Gamma^{x:\Omega}_{\hat\nu}\vdash \mathbf{W}_4:\Omega}\,(\mathrm{Ax}\text{-}\Omega)}{\Gamma^{x:\Omega}_{\hat\nu}\vdash y(\mathbf{W}_4):\hat\mu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.y(\mathbf{W}_4):\Omega\to\hat\mu}\,(\to\mathrm{I})}{(\mathsf{D3})\quad \Gamma_{\hat\nu}\vdash\lambda x.y(\mathbf{W}_4):\hat\mu}\,(\leq) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x:\hat\mu}\,(\mathrm{Ax})}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x:\Omega\to\mu}\,(\leq) \quad \dfrac{}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash \mathbf{W}_4:\Omega}\,(\mathrm{Ax}\text{-}\Omega)}{\Gamma^{x:\hat\mu}_{\hat\nu}\vdash x(\mathbf{W}_4):\mu}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.x(\mathbf{W}_4):\hat\mu\to\mu}\,(\to\mathrm{I})}{(\mathsf{D4})\quad \Gamma_{\hat\nu}\vdash\lambda x.x(\mathbf{W}_4):\mu}\,(\leq)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma}\,(\mathrm{Ax})}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma\to\gamma}\,(\leq) \quad \dfrac{}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma}\,(\mathrm{Ax})}{\Gamma^{x:\gamma}_{\hat\nu}\vdash xx:\gamma}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.xx:\gamma\to\gamma}\,(\to\mathrm{I}) \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma}\,(\mathrm{Ax})}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma\to\gamma}\,(\leq) \quad \dfrac{}{\Gamma^{x:\gamma}_{\hat\nu}\vdash x:\gamma}\,(\mathrm{Ax})}{\Gamma^{x:\gamma}_{\hat\nu}\vdash xx:\gamma}\,(\to\mathrm{E})}{\Gamma_{\hat\nu}\vdash\lambda x.xx:\gamma\to\gamma}\,(\to\mathrm{I})}{\Gamma_{\hat\nu}\vdash\lambda x.xx:\gamma}\,(\leq)}{(\mathsf{D5})\quad \Gamma_{\hat\nu}\vdash(\lambda x.xx)(\lambda x.xx):\gamma}\,(\to\mathrm{E})$$

Fig. 4. Type derivations

(4) $\Gamma \vdash^{\nabla} \lambda x.\mathbf{t} : B \rightarrow A$ *iff* $\Gamma, x : B \vdash^{\nabla} \mathbf{t} : A$.

Note that in the first case of the theorem, we have to suppose that $A \not\sim^{\nabla} \Omega$, since we can derive $\vdash^{\nabla} x : \Omega$ using axiom $(\text{Ax-}\Omega)$.

The main motivation for introducing the type assignment systems is to get the meaning of a λ-term in the inverse limit λ-models by means of the types which are deducible for it.

Recall that mappings $\rho : \mathtt{var} \mapsto \mathcal{F}^{\mathcal{D}}$ and $\rho : \mathtt{var} \mapsto \mathcal{F}^{\mathcal{E}}$ are called environments. The notation $\Gamma \triangleright \rho$ means that for $(x : B) \in \Gamma$ one has that $B \in \rho(x)$. The proof of the following theorem by induction on λ-terms using the Generation Theorem 25 is easy.

Theorem 26 (Finitary logical descriptions) *(1) For any λ-term $\mathbf{t}$ and environment $\rho : \mathtt{var} \mapsto \mathcal{F}^{\mathcal{D}}$,*

$$[\![\mathbf{t}]\!]^{\mathcal{F}^{\mathcal{D}}}_{\rho} = \{A \in \mathbb{T}^{\mathcal{D}} \mid \exists \Gamma.\ \Gamma \triangleright \rho \ \&\ \Gamma \vdash^{\mathcal{D}} \mathbf{t} : A\};$$

(2) For any λ-term $\mathbf{t}$ and environment $\rho : \mathtt{var} \mapsto \mathcal{F}^{\mathcal{E}}$,

$$[\![\mathbf{t}]\!]^{\mathcal{F}^{\mathcal{E}}}_{\rho} = \{A \in \mathbb{T}^{\mathcal{E}} \mid \exists \Gamma.\ \Gamma \triangleright \rho \ \&\ \Gamma \vdash^{\mathcal{E}} \mathbf{t} : A\}.$$

As an immediate consequence we get that typings are invariant under subject conversion.

Corollary 27 *If $\Gamma \vdash_{\nabla} \mathbf{t} : A$ and $\mathbf{t} =_{\beta} \mathbf{u}$, then $\Gamma \vdash_{\nabla} \mathbf{u} : A$.*

Theorems 19 and 26 allow us to rephrase the main theorem of Section 3, Theorem 9, as follows:

Theorem 28 (Main Theorem, Version II)

(1) $\mathbf{t} \in \mathcal{PN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \hat{\nu}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \hat{\nu}$;
(2) $\mathbf{t} \in \mathcal{N}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \nu$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \nu$;
(3) $\mathbf{t} \in \mathcal{PHN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \hat{\mu}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \hat{\mu}$;
(4) $\mathbf{t} \in \mathcal{HN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \mu$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \mu$;
(5) $\mathbf{t} \in \mathcal{PWN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \Omega^{n} \rightarrow \Omega$ *for all* $n \in \mathbb{N}$;
(6) $\mathbf{t} \in \mathcal{WN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{D}} \mathbf{t} : \Omega \rightarrow \Omega$;
(7) $\mathbf{t} \in \mathcal{CN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \gamma \cap \nu$;
(8) $\mathbf{t} \in \mathcal{CHN}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \gamma \cap \mu$;
(9) $\mathbf{t} \in \mathcal{C}$ *iff* $\Gamma_{\hat{\nu}} \vdash^{\mathcal{E}} \mathbf{t} : \gamma$.

The proofs of the (⇒) parts of this Theorem are mainly straightforward inductions and case split, with the exception of the case of persistently normalising terms, which is treated in Section 6. The proofs of the (⇐) parts require the set-theoretic semantics of intersection types using saturated sets, which is developed in Section 7.

Proof of Theorem 28*(9)-(2)***($\Rightarrow$)**

In this proof we will use the characterisations of $\mathcal{PWN}$ and $\mathcal{PHN}$ given in the proof of Proposition 6.

(9) We will show $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{t} : \gamma$ for all $\mathbf{t}$ by structural induction on $\mathbf{t}$. If $\mathbf{t}$ is a variable it is trivial. If $\mathbf{t} \equiv \mathbf{up}$, then by the induction hypothesis $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{u} : \gamma$ and $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{p} : \gamma$. By rule $(\leq_{\mathcal{E}})$ we get $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{u} : \gamma \to \gamma$ and therefore using $(\to \mathrm{E})$ we conclude $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{t} : \gamma$. If $\mathbf{t} \equiv \lambda x.\mathbf{u}$, by the induction hypothesis $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{u} : \gamma$. Then using $(\to \mathrm{I})$ we deduce $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{t} : \gamma \to \gamma$, and we conclude by $(\leq_{\mathcal{E}})$ $\Gamma_\gamma \vdash^{\mathcal{E}} \mathbf{t} : \gamma$. We can conclude $\vdash^{\mathcal{E}} \mathbf{t} : \gamma$ for all closed $\mathbf{t}$, and by rule (W) $\Gamma_{\hat\nu} \vdash^{\mathcal{E}} \mathbf{t} : \gamma$. Derivation (D5) in Figure 4 is a paradigmatic example.

(6) By Corollary 27 it suffices to consider $\mathbf{t}$ in weak head normal form. If $\mathbf{t} \equiv \lambda x.\mathbf{u}$, then we get $\Gamma_{\hat\nu}^{x:\Omega} \vdash^{\mathcal{D}} \mathbf{u} : \Omega$ by (Ax-Ω) and $\Gamma_{\hat\nu} \vdash^{\mathcal{D}} \mathbf{t} : \Omega \to \Omega$ by rule $(\to \mathrm{I})$. If $\mathbf{t} \equiv x\vec{\mathbf{u}}$, where m is the length of $\vec{\mathbf{u}}$, being $\hat\nu \leq_{\mathcal{D}} \Omega^{m+1} \to \Omega$, we derive $\Gamma_{\hat\nu} \vdash^{\mathcal{D}} \mathbf{t} : \Omega \to \Omega$ using (Ax-Ω), $(\leq_{\mathcal{D}})$ and $(\to \mathrm{E})$.

(5) If $\mathbf{t}$ is an unsolvable term of order ∞, i.e. for all $n \geq 0$, there is $\mathbf{u}$ such that $\mathbf{t} =_\beta \lambda x_1 \ldots x_n.\mathbf{u}$, we can derive $\Gamma_{\hat\nu} \vdash^{\mathcal{D}} \lambda x_1 \ldots x_n.\mathbf{u} : \Omega^n \to \Omega$ by (Ax-Ω) and rule $(\to \mathrm{I})$. If $\mathbf{t}$ is a solvable term such that the head variable of its head normal form is free, i.e. $\mathbf{t} =_\beta \lambda\vec{x}.y\vec{\mathbf{u}}$, being $\hat\nu \leq_{\mathcal{D}} \Omega^{m+l} \to \Omega$ for all l, we can derive $\Gamma_{\hat\nu} \vdash^{\mathcal{D}} \lambda\vec{x}.y\vec{\mathbf{u}} : \Omega^{n+l} \to \Omega$, where m is the length of $\vec{\mathbf{u}}$ and n is the length of $\vec{x}$.

(4) Again by Corollary 27 it suffices to consider $\mathbf{t}$ in head normal form. Let $\mathbf{t} \equiv \lambda\vec{y}.x\vec{\mathbf{u}}$ where $\vec{y}$ has length n and $\vec{\mathbf{u}}$ has length m. We have $\Gamma_{\hat\mu} \vdash^{\nabla} x\vec{\mathbf{u}} : \mu$ by rules (Ax-Ω), $(\leq_\nabla)$ and $(\to \mathrm{E})$ being $\hat\mu \leq_\nabla \Omega^m \to \mu$. By $(\to \mathrm{I})$ this implies $\Gamma_{\hat\mu} \vdash^{\nabla} \mathbf{t} : \hat\mu^n \to \mu$. We conclude $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \mu$ using $(\leq_\nabla)$ and $(\leq_\nabla \mathrm{L})$. An example is derivation (D4) in Figure 4.

(8) follows from *(4)* and *(9)* being $\mathcal{CHN} = \mathcal{C} \cap \mathcal{HN}$.

(3) The head variable of the head normal form of $\mathbf{t}$ must be free. We can type a term of the shape $\lambda\vec{x}.y\vec{\mathbf{u}}$, where $y \notin \vec{x}$ as follows: $\Gamma_{\hat\nu} \vdash^{\nabla} \lambda\vec{x}.y\vec{\mathbf{u}} : \Omega^n \to \hat\mu$, since $\hat\nu \leq_\nabla \Omega^m \to \hat\mu$, where m is the length of $\vec{\mathbf{u}}$ and n is the length of $\vec{x}$. We conclude $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \hat\mu$ using $(\leq_\nabla)$. See the derivation (D3) in Figure 4 as an example.

(2) By *(4)*, we get $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \mu$. Since $\mu \sim \hat\mu \to \mu$, we only need to prove $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \hat\nu \to \nu$. The proof is by induction on the normal form $\mathbf{t}$. If $\mathbf{t}$ is a variable it is trivial, since $\hat\nu \leq_\nabla \hat\nu \to \nu$. If $\mathbf{t} \equiv x\vec{\mathbf{u}}$ then by induction $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{u} : \nu$ for all $\mathbf{u} \in \vec{\mathbf{u}}$ and we get $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \hat\nu \to \nu$ since $\hat\nu \leq_\nabla \nu^m \to \hat\nu \to \nu$, where m is the length of $\vec{\mathbf{u}}$. If $\mathbf{t} \equiv \lambda x.\mathbf{u}$ then by induction $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{u} : \nu$ and this gives $\Gamma_{\hat\nu} \vdash^{\nabla} \mathbf{t} : \hat\nu \to \nu$ by rule $(\to \mathrm{I})$. A paradigmatic example is derivation (D2) in Figure 4.

(7) follows from *(2)* and *(9)* being $\mathcal{CN} = \mathcal{C} \cap \mathcal{N}$.

6 Persistently Normalising Terms

In the following section we omit the index ∇. We need some definitions in order to state the characterisation of the set $\mathcal{PN}$ given in Böhm and Dezani [8] (Theorem 34).

We split the occurrences of variables in normal forms into replaceable and non-replaceable ones. Roughly a variable is replaceable in $\mathbf{t}$ iff it can be replaced by an arbitrary term when $\mathbf{t}$ is applied to a suitable sequence of arguments.

Definition 29 (Replaceable and non-replaceable variables) *In a normal form* $\mathbf{t}$*:*

- *All occurrences of variables bound in the initial abstractions are* replaceable.
- *All occurrences of free variables are* non-replaceable.
- *Let* $x\vec{\mathbf{u}}(\lambda\vec{z}y.\mathbf{p})$ *be a subterm of* $\mathbf{t}$*. Then the occurrences of* y *in* $\mathbf{p}$ *are* (non)-replaceable *if the showed occurrence of* x *is* (non)-replaceable.

Let $\mathtt{RV}(\mathbf{t})$ *and* $\mathtt{NV}(\mathbf{t})$ *denote, respectively, the sets of replaceable and non-replaceable variables of* $\mathbf{t}$*.*

Example 30 *In the term* $\lambda x.y(\lambda z.x(yx)(\lambda u.zu))$ *the variables* x, u *are replaceable and the variables* y, z *are non-replaceable.*

It is easy to verify that each occurrence of a variable in a normal form is either replaceable or non-replaceable according to the previous definition.

We can classify the subterms of a normal form, which are arguments of a variable, according to whether the occurrence of this variable is replaceable or non-replaceable. Informally, a subterm of a normal form $\mathbf{t}$ is safe when it remains an argument of the same variable when $\mathbf{t}$ is applied to arbitrary terms.

Definition 31 (Safe and unsafe subterms) *Let* $\mathbf{t}$ *be a normal form and* $x\vec{\mathbf{p}}\mathbf{u}$ *be a subterm of* $\mathbf{t}$*. Then (the showed occurrence of)* $\mathbf{u}$ *is:*

- *an* unsafe *subterm if (the showed occurrence of)* x *is replaceable;*
- *a* safe *subterm if (the showed occurrence of)* x *is non-replaceable.*

Moreover $\mathbf{t}$ *is an* unsafe *subterm of* $\mathbf{t}$*.*

Example 32 *In the term* $\lambda x.y(\lambda z.x(yx)(\lambda u.zu))$ *the term itself and the proper subterms* yx *and* $\lambda u.zu$ *are unsafe while the subterms* $\lambda z.x(yx)(\lambda u.zu), u$ *are safe.*

Remark 33 *The variables bound in the initial abstractions of a safe subterm are non-replaceable, while the variables bound in the initial abstractions of an unsafe* *subterm are replaceable.*

Theorem 34 (Characterisation of $\mathcal{PN}$) *(Böhm and Dezani [8]) A term is persistently normalising iff in its normal form all head variables of unsafe subterms are non-replaceable.*

Example 35 *In the term $\lambda x.y(\lambda z.x(yx)(\lambda u.zu))$ the unsafe subterms are the term itself and the proper subterms $yx, \lambda u.zu$: the variables y, z are non-replaceable, therefore $\lambda x.y(\lambda z.x(yx)(\lambda u.zu)) \in \mathcal{PN}$.*

In particular it follows that if $\mathbf{t} \in \mathcal{PN}$ is a normal form, then its head variable is free.

We will use this characterisation in order to prove that if $\mathbf{t} \in \mathcal{PN}$ then $\Gamma_{\hat{\nu}} \vdash \mathbf{t} : \hat{\nu}$. Before that we give the last definition and three lemmas about the types of safe and unsafe subterms.

Definition 36 *Let $\mathbf{t}$ be a normal form, $\mathbf{u}$ be a subterm of $\mathbf{t}$ and A, B be types. We define the basis $\Sigma(\mathbf{t}, \mathbf{u}, A, B)$ as follows:*

$$\Sigma(\mathbf{t}, \mathbf{u}, A, B) = \{x : A \mid x \in \mathtt{RV}(\mathbf{t}) \cap \mathsf{FV}(\mathbf{u})\} \cup \{x : B \mid x \in \mathtt{NV}(\mathbf{t}) \cap \mathsf{FV}(\mathbf{u})\}.$$

Lemma 37 *Let $\mathbf{t} \in \mathcal{PN}$ be a normal form, and $\lambda\vec{x}.\mathbf{u}$ be an unsafe subterm of $\mathbf{t}$. Then:*

$$\Sigma(\mathbf{t}, \mathbf{u}, \Omega, \hat{\mu}) \vdash \mathbf{u} : \hat{\mu}.$$

PROOF. Let $\mathbf{u} \equiv \lambda\vec{z}.y\vec{\mathbf{p}}$ and $\Gamma = \Sigma(\mathbf{t}, \mathbf{u}, \Omega, \hat{\mu})$. By Theorem 34 y is non-replaceable in $\mathbf{t}$, so y is free in $\mathbf{u}$ (Remark 33), since the variables in $\vec{z}$ are replaceable by Definitions 29 and 31. Therefore $y : \hat{\mu} \in \Gamma$. Using axiom (Ax-Ω), and rules ($\leq$), ($\to$ E) we can derive $\Gamma \vdash y\vec{\mathbf{p}} : \hat{\mu}$ since $\hat{\mu} \sim \Omega \to \hat{\mu}$. We can conclude by rules ($\to$ I) and ($\leq$) taking as premises $z : \Omega$ for all $z \in \vec{z}$.

Lemma 38 *Let $\mathbf{t} \in \mathcal{PN}$ be a normal form, and $\lambda\vec{x}.\mathbf{u}$ be a safe subterm of $\mathbf{t}$. Then:*

$$\Sigma(\mathbf{t}, \mathbf{u}, \mu, \hat{\mu}) \vdash \mathbf{u} : \mu.$$

PROOF. Let $\mathbf{u} \equiv \lambda\vec{z}.y\vec{\mathbf{p}}$, $\Gamma = \Sigma(\mathbf{t}, \mathbf{u}, \mu, \hat{\mu})$ and $\Gamma' = \Gamma \cup \{z : \hat{\mu} \mid z \in \vec{z}\}$. The variables in $\vec{z}$ are non-replaceable by Definitions 29 and 31. This implies $\Sigma(\mathbf{t}, \mathbf{p}, \mu, \hat{\mu}) \subseteq \Gamma'$ for all $\mathbf{p} \in \vec{\mathbf{p}}$.
If y is replaceable, then by Definition 31 all $\mathbf{p} \in \vec{\mathbf{p}}$ are unsafe subterms, so by Lemma 37 $\Sigma(\mathbf{t}, \mathbf{p}, \Omega, \hat{\mu}) \vdash \mathbf{p} : \hat{\mu}$ for all $\mathbf{p} \in \vec{\mathbf{p}}$. By rules ($\leq$ L) and (W) we get $\Gamma' \vdash \mathbf{p} : \hat{\mu}$ for all $\mathbf{p} \in \vec{\mathbf{p}}$. From $y : \mu \in \Gamma'$ we can then derive $\Gamma' \vdash y\vec{\mathbf{p}} : \mu$ using rules ($\leq$), and ($\to$ E), since $\mu \sim \hat{\mu} \to \mu$. We can conclude by rules ($\to$ I) and ($\leq$).
If y is non-replaceable, from $y : \hat{\mu} \in \Gamma'$ using axiom (Ax-Ω), and rules ($\leq$), ($\to$ E)

we can derive $\Gamma \vdash y\vec{\mathbf{p}} : \hat{\mu}$, since $\hat{\mu} \sim \Omega \to \hat{\mu}$. By rule $(\leq)$ this gives $\Gamma \vdash y\vec{\mathbf{p}} : \mu$, so we can conclude using rules $(\to \mathrm{I})$ and $(\leq)$.

Lemma 39 *Let* $\mathbf{t} \in \mathcal{PN}$ *be a normal form. Then:*

$$\Sigma(\mathbf{t}, \mathbf{u}, \nu, \hat{\nu}) \vdash \mathbf{u} : \hat{\nu} \text{ whenever } \lambda\vec{x}.\mathbf{u} \text{ is an unsafe subterm of } \mathbf{t};$$
$$\Sigma(\mathbf{t}, \mathbf{u}, \nu, \hat{\nu}) \vdash \mathbf{u} : \nu \text{ whenever } \lambda\vec{x}.\mathbf{u} \text{ is a safe subterm of } \mathbf{t}.$$

PROOF. The proof is by structural induction on safe and unsafe subterms of $\mathbf{t}$. The first step, i.e. when $\mathbf{u}$ is a variable, is trivial.
For the induction step let $\Gamma = \Sigma(\mathbf{t}, \mathbf{u}, \nu, \hat{\nu})$. We distinguish the following cases.
If $\mathbf{u} \equiv y\vec{\mathbf{p}}$ and $\lambda\vec{x}.\mathbf{u}$ is unsafe, then by Theorem 34 y is non-replaceable in $\mathbf{t}$, so $y : \hat{\nu} \in \Gamma$. By Definition 31 all $\mathbf{p} \in \vec{\mathbf{p}}$ are safe subterms, and so by the induction hypothesis $\Sigma(\mathbf{t}, \mathbf{p}, \nu, \hat{\nu}) \vdash \mathbf{p} : \nu$. By rule (W) we get $\Gamma \vdash \mathbf{p} : \nu$ for all $\mathbf{p} \in \vec{\mathbf{p}}$, so we conclude $\Gamma \vdash \mathbf{u} : \hat{\nu}$ using $(\leq)$ and $(\to \mathrm{E})$, since $\hat{\nu} \leq \nu \to \hat{\nu}$.
If $\mathbf{u} \equiv y\vec{\mathbf{p}}$ and $\lambda\vec{x}.\mathbf{u}$ is safe, then y can be either replaceable or non-replaceable, but in both cases $\Gamma \vdash y : \nu$. By Definition 31 all $\mathbf{p} \in \vec{\mathbf{p}}$ are unsafe subterms, so by the induction hypothesis $\Sigma(\mathbf{t}, \mathbf{p}, \nu, \hat{\nu}) \vdash \mathbf{p} : \hat{\nu}$. We conclude as in the previous case since $\nu \leq \hat{\nu} \to \nu$.
If $\mathbf{u} \equiv \lambda z.\mathbf{u}'$ and $\lambda\vec{x}.\mathbf{u}$ is unsafe, then by the induction hypothesis $\Sigma(\mathbf{t}, \mathbf{u}', \nu, \hat{\nu}) \vdash \mathbf{u}' : \hat{\nu}$. By Definitions 29 and 31 z is replaceable in $\mathbf{t}$, therefore $z : \nu \in \Sigma(\mathbf{t}, \mathbf{u}', \nu, \hat{\nu})$. Using rule $(\to \mathrm{I})$, we get $\Gamma \vdash \mathbf{u} : \nu \to \hat{\nu}$. By Lemma 37 $\Sigma(\mathbf{t}, \mathbf{u}, \Omega, \hat{\mu}) \vdash \mathbf{u} : \hat{\mu}$. Using rules $(\leq \mathrm{L})$ and $(\cap\mathrm{I})$ we get $\Gamma \vdash \mathbf{u} : \hat{\mu} \cap (\nu \to \hat{\nu})$, so we can conclude by $(\leq)$, since $\hat{\nu} \sim \hat{\mu} \cap (\nu \to \hat{\nu})$.
If $\mathbf{u} \equiv \lambda z.\mathbf{u}'$ and $\lambda\vec{x}.\mathbf{u}$ is safe, the proof is similar to the previous case using Lemma 38 instead of Lemma 37.

Proof of Theorem 28*(1)***(⇒)** By Corollary 27 it suffices to consider $\mathbf{t}$ in normal form, and Lemma 39 yields the conclusion as $\mathbf{t}$ is an unsafe term itself, using rules $(\leq \mathrm{L})$ and (W). Derivation (D1) in Figure 4 is a paradigmatic example.

7 Reducibility Method

In order to prove the (⇐)-part of our main statement (Theorem 28), we will use the set theoretic semantics of intersection types and saturated sets, which is referred to as the reducibility method.

The *reducibility method* was introduced by Tait [32] for proving the strong normalisation property of simply typed λ-calculus. Further it was developed in Tait [33] and Girard [17] for proving the strong normalisation property of polymorphic λ-calculus.

In Pottinger [27], van Bakel [34], Krivine [20], [21], Ghilezan [15], Amadio and Curien [5], the reducibility method is applied in order to characterise all and only the strongly normalising λ-terms in λ-calculus with intersection types. The reducibility method is also used for characterising some special classes of λ-terms such as strongly normalising terms, normalising terms, head normalising terms, and weak head normalising terms. They are characterised by their typeability in various intersection type assignment systems in Leivant [22] and Gallier [14], whereas both the mentioned terms as well as their persistent versions are characterised in Dezani et al. [13]. Furthermore, this method was applied in the proof of the Church-Rosser property (confluence) of the simply typed λ-calculus in Statman [31], Koletsos [19], and Mitchell [24], [25].

We will adapt the reducibility method, by requiring that the terms typeable with the key types listed in Theorem 28 belong to the corresponding sets.

In order to develop the reducibility method we consider Λ as the *applicative structure* whose domain are λ-terms and where the application is just the application of terms.

We first define the *interpretations of types* in $\mathbb{T}^{\mathcal{D}}$ and in $\mathbb{T}^{\mathcal{E}}$: the only difference between the two interpretations concerns the arrow constructor. Let $\mathcal{P}(\Lambda)$ denote the powerset of Λ.

Definition 40

(1) *The map* $[\![-]\!]^{\mathcal{D}} : \mathbb{T}^{\mathcal{D}} \to \mathcal{P}(\Lambda)$ *is defined by:*
$[\![\nu]\!]^{\mathcal{D}} = \mathcal{N}$, $[\![\hat{\nu}]\!]^{\mathcal{D}} = \mathcal{PN}$, $[\![\mu]\!]^{\mathcal{D}} = \mathcal{HN}$, $[\![\hat{\mu}]\!]^{\mathcal{D}} = \mathcal{PHN}$, $[\![\Omega]\!]^{\mathcal{D}} = \Lambda$;
$[\![A \cap B]\!]^{\mathcal{D}} = [\![A]\!]^{\mathcal{D}} \cap [\![B]\!]^{\mathcal{D}}$;
$[\![A \to B]\!]^{\mathcal{D}} = [\![A]\!]^{\mathcal{D}} \stackrel{\mathcal{D}}{\longmapsto} [\![B]\!]^{\mathcal{D}} = \{\mathbf{t} \in \mathcal{WN} \mid \forall \mathbf{u} \in [\![A]\!]^{\mathcal{D}} \quad \mathbf{tu} \in [\![B]\!]^{\mathcal{D}}\}$.

(2) *The map* $[\![-]\!]^{\mathcal{E}} : \mathbb{T}^{\mathcal{E}} \to \mathcal{P}(\Lambda)$ *is defined by:*
$[\![\nu]\!]^{\mathcal{E}} = \mathcal{N}$, $[\![\hat{\nu}]\!]^{\mathcal{E}} = \mathcal{PN}$, $[\![\mu]\!]^{\mathcal{E}} = \mathcal{HN}$ $[\![\hat{\mu}]\!]^{\mathcal{E}} = \mathcal{PHN}$, $[\![\gamma]\!]^{\mathcal{E}} = \mathcal{C}$, $[\![\Omega]\!]^{\mathcal{E}} = \Lambda$;
$[\![A \cap B]\!]^{\mathcal{E}} = [\![A]\!]^{\mathcal{E}} \cap [\![B]\!]^{\mathcal{E}}$;
$[\![A \to B]\!]^{\mathcal{E}} = [\![A]\!]^{\mathcal{E}} \stackrel{\mathcal{E}}{\longmapsto} [\![B]\!]^{\mathcal{E}} = \{\mathbf{t} \in \Lambda \mid \forall \mathbf{u} \in [\![A]\!]^{\mathcal{E}} \quad \mathbf{tu} \in [\![B]\!]^{\mathcal{E}}\}$.

Observe the relation between $\stackrel{\mathcal{D}}{\longmapsto}$ and $\stackrel{\mathcal{E}}{\longmapsto}$:

$$\mathcal{S} \stackrel{\mathcal{D}}{\longmapsto} \mathcal{T} = (\mathcal{S} \cap \mathcal{WN}) \stackrel{\mathcal{E}}{\longmapsto} \mathcal{T}.$$

Notice that

$$\begin{aligned} &[\![\Omega \to \Omega]\!]^{\mathcal{D}} = \{\mathbf{t} \in \mathcal{WN} \mid \forall \mathbf{u} \in [\![\Omega]\!]^{\mathcal{D}}\ \mathbf{tu} \in [\![\Omega]\!]^{\mathcal{D}}\} \\ &= \{\mathbf{t} \in \mathcal{WN} \mid \forall \mathbf{u} \in \Lambda\ \mathbf{tu} \in \Lambda\} = \mathcal{WN}, \end{aligned}$$

and by Definition 3

$$\begin{aligned}
&\bigcap_{n\in\mathbb{N}}[\![\Omega^n \to \Omega]\!]^{\mathcal{D}} = \bigcap_{n\in\mathbb{N}}[\![\Omega^n \to \Omega \to \Omega]\!]^{\mathcal{D}}\\
&= \{\mathbf{t} \in \mathcal{WN} \mid \forall \vec{\mathbf{u}} \in [\![\Omega]\!]^{\mathcal{D}}\ \mathbf{t}\vec{\mathbf{u}} \in [\![\Omega \to \Omega]\!]^{\mathcal{D}}\}\\
&= \{\mathbf{t} \in \mathcal{WN} \mid \forall \vec{\mathbf{u}} \in \Lambda\ \mathbf{t}\vec{\mathbf{u}} \in \mathcal{WN}\} = \mathcal{PWN}.
\end{aligned}$$

The following definition of *saturated set* is standard, see Krivine [20], [21].

Definition 41 *A set $\mathcal{S} \subseteq \Lambda$ is* saturated, *notation SAT$(\mathcal{S})$, if*

$$(\forall \mathbf{t}, \mathbf{u}, \vec{\mathbf{p}} \in \Lambda)\ \mathbf{t}[x := \mathbf{u}]\vec{\mathbf{p}} \in \mathcal{S} \ \Rightarrow\ (\lambda x.\mathbf{t})\mathbf{u}\,\vec{\mathbf{t}} \in \mathcal{S}.$$

Obviously, each of the sets $\mathcal{PN}$, $\mathcal{N}$, $\mathcal{PHN}$, $\mathcal{HN}$, $\mathcal{C}$, and Λ satisfies the above condition, since they are closed under β-conversion. We can show that both type interpretations are saturated.

Lemma 42 $(\forall A \in \mathbb{T}^{\nabla})$*SAT*$([\![A]\!]^{\nabla})$.

PROOF. The proof is by structural induction on types. The only interesting case is that of arrow types. Let $\mathbf{t}, \mathbf{u}, \vec{\mathbf{p}} \in \Lambda$. Suppose $\mathbf{t}[x := \mathbf{u}]\vec{\mathbf{p}} \in [\![A \to B]\!]^{\mathcal{D}}$. Let $\mathbf{q} \in [\![A]\!]^{\mathcal{D}}$ be arbitrary. By Definition 40(1) $\mathbf{t}[x := \mathbf{u}]\vec{\mathbf{p}}\mathbf{q} \in [\![B]\!]^{\mathcal{D}}$. Then by the induction hypothesis $(\lambda x.\mathbf{t})\mathbf{u}\vec{\mathbf{p}}\mathbf{q} \in [\![B]\!]^{\mathcal{D}}$. Moreover, by Definition 40(1) we get $\mathbf{t}[x := \mathbf{u}]\vec{\mathbf{p}} \in \mathcal{WN}$, and this implies $(\lambda x.\mathbf{t})\mathbf{u}\vec{\mathbf{p}} \in \mathcal{WN}$. Since $\mathbf{q}$ was arbitrary, according to Definition 40(1) we get $(\lambda x.\mathbf{t})\mathbf{u}\vec{\mathbf{p}} \in [\![A \to B]\!]^{\mathcal{D}}$. Similarly one can show that $\mathbf{t}[x := \mathbf{u}]\vec{\mathbf{p}} \in [\![A \to B]\!]^{\mathcal{E}}$ implies $(\lambda x.\mathbf{t})\mathbf{u}\vec{\mathbf{p}} \in [\![A \to B]\!]^{\mathcal{E}}$.

We can simplify Lemma 42.

Corollary 43 $(\forall A \in \mathbb{T}^{\nabla})$ $(\forall \mathbf{u} \in \Lambda)$ $\mathbf{t}[x := \mathbf{u}] \in [\![A]\!]^{\nabla} \Rightarrow (\lambda x.\mathbf{t})\mathbf{u} \in [\![A]\!]^{\nabla}$.

The preorders on types agree with the set theoretic inclusion between type interpretations.

Lemma 44 *If $A \leq_{\nabla} B$, then* $[\![A]\!]^{\nabla} \subseteq [\![B]\!]^{\nabla}$.

PROOF. By induction on the length of the derivation of $A \leq_{\nabla} B$. Proposition 6 justifies the axioms between atomic types. Axioms $(\hat{\mu} \to)$ and $(\mu \to)$ follow from Definitions 2 and 3. Axiom $(\hat{\nu} \to)$ follows from the same definitions taking into account that $\mathcal{PN} \subseteq \Lambda \to \mathcal{PHN}$ since $\mathcal{PHN} = \Lambda \to \mathcal{PHN}$. Axiom $(\nu \to)$ follows from Proposition 7 taking into account that $\mathcal{N} \subseteq \mathcal{PHN} \to \mathcal{HN}$ since $\mathcal{HN} = \mathcal{PHN} \to \mathcal{HN}$.

We define the ∇-*valuations of terms* $[\![-]\!]^\nabla_\theta : \Lambda \to \Lambda$ and the *semantic satisfiability relations* $\models^\nabla$ which connect the type interpretations and the term valuations as follows.

Definition 45 *Let $[\![-]\!]^\nabla : \mathbb{T}^\nabla \to \mathcal{P}(\Lambda)$, $\nabla \in \{\mathcal{D}, \mathcal{E}\}$, be the defined type interpretation and let $\theta : \mathtt{var} \to \Lambda$ be a valuation of term variables in Λ. Then*

(1) $[\![-]\!]^\nabla_\theta : \Lambda \to \Lambda$ *is defined by*
$[\![\mathbf{t}]\!]^\nabla_\theta = \mathbf{t}[x_1 := \theta(x_1), \ldots, x_n := \theta(x_n)]$, *where* $\mathsf{FV}(\mathbf{t}) = \{x_1, \ldots, x_n\}$*;*
(2) $\theta \models^\nabla \mathbf{t} : A$ *iff* $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![A]\!]^\nabla$*;*
(3) $\theta \models^\nabla \Gamma$ *iff* $(\forall (x : A) \in \Gamma)$ $\theta \models^\nabla x : A$*;*
(4) $\Gamma \models^\nabla \mathbf{t} : A$ *iff* $(\forall \theta \models^\nabla \Gamma)$ $\theta \models^\nabla \mathbf{t} : A$*.*

We can prove that our type assignment systems are *sound* for the above semantic satisfiability.

Theorem 46 (Soundness)

$$\Gamma \vdash^\mathcal{D} \mathbf{t} : A \Rightarrow \Gamma \models^\mathcal{D} \mathbf{t} : A \qquad \Gamma \vdash^\mathcal{E} \mathbf{t} : A \Rightarrow \Gamma \models^\mathcal{E} \mathbf{t} : A.$$

PROOF.

By induction on the derivation of $\Gamma \vdash^\nabla \mathbf{t} : A$.

Case 1. The last step is (Ax), i.e. $\Gamma, x : A \vdash^\nabla x : A$. Then $\Gamma, x : A \models^\nabla x : A$ by Definition 45*(3)*.

Case 2. The last step is ($\to$ E), i.e. $\Gamma \vdash^\nabla \mathbf{t} : A \to B, \Gamma \vdash^\nabla \mathbf{u} : A \Rightarrow \Gamma \vdash^\nabla \mathbf{tu} : B$. Then by the induction hypothesis $\Gamma \models^\nabla \mathbf{t} : A \to B$ and $\Gamma \models^\nabla \mathbf{u} : A$. Let $\theta \models^\nabla \Gamma$, then $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![A \to B]\!]^\nabla = [\![A]\!]^\nabla \overset{\nabla}{\longmapsto} [\![B]\!]^\nabla$ and $[\![\mathbf{u}]\!]^\nabla_\theta \in [\![A]\!]^\nabla$. Therefore $[\![\mathbf{tu}]\!]^\nabla_\theta \equiv [\![\mathbf{t}]\!]^\nabla_\theta [\![\mathbf{u}]\!]^\nabla_\theta \in [\![B]\!]^\nabla$.

Case 3. The last step is ($\to$ I), i.e. $\Gamma, x : A \vdash^\nabla \mathbf{t} : B \Rightarrow \Gamma \vdash^\nabla \lambda x.\mathbf{t} : A \to B$. By the induction hypothesis $\Gamma, x : A \models^\nabla \mathbf{t} : B$. Let $\theta \models^\nabla \Gamma$ and let $\mathbf{u} \in [\![A]\!]^\nabla$. We define $\theta[x := \mathbf{u}](x) = \mathbf{u}, \theta[x := \mathbf{u}](y) = \theta(y)$ for $x \neq y$. Then $\theta[x := \mathbf{u}] \models^\nabla \Gamma$, since $x \notin \Gamma$, and $\theta[x := \mathbf{u}] \models^\nabla x : A$, since $\mathbf{u} \in [\![A]\!]^\nabla$. Therefore $\theta[x := \mathbf{u}] \models^\nabla \mathbf{t} : B$, i.e. $[\![\mathbf{t}]\!]^\nabla_{\theta[x := \mathbf{u}]} \in [\![B]\!]^\nabla$, which means by Definition 45*(1)* that $\mathbf{t}[\vec{y} := \theta(\vec{y})][x := \mathbf{u}] \in [\![B]\!]^\nabla$, where $\vec{y} = \mathsf{FV}(\mathbf{t}) \setminus \{x\}$. By Corollary 43 we have $(\lambda x.\mathbf{t}[\vec{y} := \theta(\vec{y})])\mathbf{u} \in [\![B]\!]^\nabla$. Then $[\![\lambda x.\mathbf{t}]\!]^\nabla_\theta \mathbf{u} \in [\![B]\!]^\nabla$ since $x \notin \mathsf{FV}(\lambda x.\mathbf{t})$. Notice that $[\![\lambda x.\mathbf{t}]\!]^\mathcal{D}_\theta \in \mathcal{WN}$. Therefore, $[\![\lambda x.\mathbf{t}]\!]^\mathcal{D}_\theta \in [\![A]\!]^\mathcal{D} \overset{\mathcal{D}}{\longmapsto} [\![B]\!]^\mathcal{D} = [\![A \to B]\!]^\mathcal{D}$, since $\mathbf{u} \in [\![A]\!]^\mathcal{D}$ was arbitrary. Similarly $[\![\lambda x.\mathbf{t}]\!]^\mathcal{E}_\theta \in [\![A \to B]\!]^\mathcal{E}$.

Case 4. The last step is ($\cap$I), i.e. $\Gamma \vdash^\nabla \mathbf{t} : A,\ \Gamma \vdash^\nabla \mathbf{t} : B \Rightarrow \Gamma \vdash^\nabla \mathbf{t} : A \cap B$. Then by the induction hypothesis $\Gamma \models^\nabla \mathbf{t} : A$ and $\Gamma \models^\nabla \mathbf{t} : B$. Let $\theta \models^\nabla \Gamma$, then $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![A]\!]^\nabla$ and $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![B]\!]^\nabla$. Therefore $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![A \cap B]\!]^\nabla$, i.e. $\Gamma \models^\nabla \mathbf{t} : A \cap B$.

Case 5. The last step is $(\leq_\nabla)$, i.e. $\Gamma \vdash^\nabla \mathbf{t} : A,\ A \leq_\nabla B \Rightarrow \Gamma \vdash^\nabla \mathbf{t} : B$. By the induction hypothesis $\Gamma \models^\nabla \mathbf{t} : A$. Let $\theta \models^\nabla \Gamma$, then $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![A]\!]^\nabla$. According to Lemma 44 $[\![A]\!]^\nabla \subseteq [\![B]\!]^\nabla$ so it follows that $[\![\mathbf{t}]\!]^\nabla_\theta \in [\![B]\!]^\nabla$, i.e. $\Gamma \models^\nabla \mathbf{t} : B$.

Proof of Theorem 28($\Leftarrow$) The proofs of all parts are similar, so we only consider part (5). Let $\Gamma_{\hat{\nu}} \vdash \mathbf{t} : \Omega^n \to \Omega$, for all n. By soundness (Theorem 46) we have that if $\theta \models^{\mathcal{D}} \Gamma_{\hat{\nu}}$, then $[\![\mathbf{t}]\!]^{\mathcal{D}}_\theta \in [\![\Omega^n \to \Omega]\!]^{\mathcal{D}}$, for all n. We can take $\theta_1(x) = x$, being $\theta_1 \models^{\mathcal{D}} \Gamma_{\hat{\nu}}$, because all variables belong to $\mathcal{PN}$. Obviously, $\theta_1(\mathbf{t}) = \mathbf{t}$ for every λ-term $\mathbf{t}$. Therefore we get that $\mathbf{t} \in [\![\Omega^n \to \Omega]\!]^{\mathcal{D}}$, for all n. Hence, $\mathbf{t} \in \mathcal{PWN}$ since $\bigcap_{n\in\mathbb{N}}[\![\Omega^n \to \Omega]\!]^{\mathcal{D}} = \mathcal{PWN}$ by Definition 40.

Remark 47 *Observe that the interpretation of terms we use in the proof of Theorem 28($\Leftarrow$) is just the identity.*

The present section is another witness of the power and elegance of the reducibility method, which allows to split the necessary double induction on types and deductions in simple statements with easy proofs.

8 Conclusion

The main contribution of the present paper is to show that two models can characterise many different sets of terms. On the one hand it seems that we cannot find elements representing weak head normalisability and closability in the same model, since the first property requires the lifting of the space of functions and this does not agree with the second one. On the other hand, there are properties which appear strongly connected, like each normalisation property with its persistent version. It is not clear if these properties can be characterised separately, i.e. if one can build models in which only one of these properties is characterised.

Acknowledgements

First of all, the authors are grateful to the referees of TYPES 02 submission and of the present submission for careful reading and many useful remarks. The final version of this paper strongly improved due to their suggestions. In particular both Section 2 of the present paper and Section 6 of [4] were added to answer referee's right criticisms. Moreover the presentation has been clarified and many proofs have been simplified. They would also like to thank Fabio Alessi for valuable collaboration in the final preparation of the present paper, especially in writing of Section 2.

References

[1] S. Abramsky. Domain theory in logical form. *Ann. Pure Appl. Logic*, 51(1-2):1–77, 1991.

[2] S. Abramsky and C.-H. L. Ong. Full abstraction in the lazy lambda calculus. *Inform. and Comput.*, 105(2):159–267, 1993.

[3] F. Alessi. *Strutture di tipi, teoria dei domini e modelli del lambda calcolo*. PhD thesis, Torino University, 1991.

[4] F. Alessi, M. Dezani-Ciancaglini, and S. Lusin. Intersection types and domain operators. *Theoret. Comput. Sci.*, 2003. This issue.

[5] R. M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*. Cambridge University Press, Cambridge, 1998.

[6] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *J. Symbolic Logic*, 48(4):931–940 (1984), 1983.

[7] H. P. Barendregt. *The Lambda Calculus: its Syntax and Semantics*. North-Holland, Amsterdam, revised edition, 1984.

[8] C. Böhm and M. Dezani-Ciancaglini. λ-terms as total or partial functions on normal forms. In C. Böhm, editor, *λ-calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 96–121, Berlin, 1975. Springer-Verlag.

[9] M. Coppo, M. Dezani, and G. Longo. Applicative information systems. In G.Ausiello and M.Protasi, editors, *CAAP'83, Trees in Algebra and Programming*, pages 35–64. Springer-Verlag, Berlin, 1983.

[10] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, and G. Longo. Extended type structures and filter lambda models. In G.Lolli, G.Longo, and A.Marcja, editors, *Logic Colloquium '82*, pages 241–262, Amsterdam, 1984. North-Holland.

[11] M. Dezani-Ciancaglini and S. Ghilezan. A lambda model characterizing computational behaviours of terms. In Y.Toyama, editor, *International Workshop on Rewriting in Proof and Computation*, pages 100–119, 2001.

[12] M. Dezani-Ciancaglini and S. Ghilezan. Two behavioural lambda models. In H.Geuvers and F.Wiedijk, editors, *Types for Proofs and Programs*, volume 2646 of *Lecture Notes in Computer Science*, pages 127–147, Berlin, 2003. Springer-Verlag.

[13] M. Dezani-Ciancaglini, F. Honsell, and Y. Motohama. Compositional characterization of λ-terms using intersection types. In M.Nielsen and B.Rovan, editors, *Mathematical Foundations of Computer Science 2000*, volume 1893 of *Lecture Notes in Computer Science*, pages 304–313, Berlin, 2000. Springer-Verlag.

[14] J. Gallier. Typing untyped λ-terms, or reducibility strikes again! *Ann. Pure Appl. Logic*, 91:231–270, 1998.

[15] S. Ghilezan. Strong normalization and typability with intersection types. *Notre Dame J. Formal Logic*, 37(1):44–52, 1996.

[16] G. K. Gierz, K. H. Hofmann, K. Keimel, L. J. D., M. W. Mislove, and D. S.

Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, 1980.

[17] J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse, et son application à l'elimination des coupures dans l'analyse et la théorie des types. In J.E.Fenstadt, editor, *2nd Scandinavian Logic Symposium*, pages 63–92, Amsterdam, 1971. North-Holland.

[18] P. T. Johnstone. *Stone Spaces*. Cambridge University Press, Cambridge, 1986. Reprint of the 1982 edition.

[19] G. Koletsos. Church-Rosser theorem for typed functionals. *J. of Symbolic Logic*, 50:782–790, 1985.

[20] J.-L. Krivine. *Lambda-calcul, types et modèles*. Masson, Paris, 1990.

[21] J.-L. Krivine. *Lambda-calculus, types and models*. Ellis Horwood, New York, 1993. Translated from the 1990 French original by René Cori.

[22] D. Leivant. Typing and computational properties of lambda expressions. *Theoret. Comput. Sci.*, 44(1):51–68, 1986.

[23] P. Martin-Löf. Lecture notes on domain interpretation of type theory. In P. Dybier, B. Nordström, K. Petersson, and J. Smith, editors, *Workshop on the Semantics of Programming Languages*. Programming Methodology Group, Chalmers University of Technology, 1983.

[24] J. C. Mitchell. Type systems for programming languages. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 415–431. Elsevier, Amsterdam, 1990.

[25] J. C. Mitchell. *Foundation for Programmimg Languages*. MIT Press, Boston, 1996.

[26] G. D. Plotkin. Set-theoretical and other elementary models of the λ-calculus. *Theoret. Comput. Sci.*, 121(1-2):351–409, 1993.

[27] G. Pottinger. A type assignment for the strongly normalizable λ-terms. In J.P.Seldin and J.R.Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 561–577. Academic Press, London, 1980.

[28] D. S. Scott. Continuous lattices. In F.W.Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136, Berlin, 1972. Springer-Verlag.

[29] D. S. Scott. Open problem. In C. Böhm, editor, *Lambda Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, page 369. Springer-Verlag, Berlin, 1975.

[30] D. S. Scott. Domains for denotational semantics. In M.Nielsen and E.M.Schmidt, editors, *Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer-Verlag, Berlin, 1982.

[31] R. Statman. Logical relations and the typed λ-calculus. *Inform. and Control*, 65:85–97, 1985.

[32] W. W. Tait. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic*, 32:198–212, 1967.

[33] W. W. Tait. A realizability interpretation of the theory of species. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *Lecture Notes in Mathematics*, pages

240–251, Berlin, 1975. Springer-Verlag.

[34] S. van Bakel. Complete restrictions of the intersection type discipline. *Theoret. Comput. Sci.*, 102(1):135–163, 1992.