Two Behavioural Lambda Models

Mariangiola Dezani-Ciancaglini[1⋆] and **Silvia Ghilezan**[2⋆⋆]

[1] Dipartimento di Informatica, Università di Torino, Torino, Italy
dezani@di.unito.it

[2] Faculty of Engineering, University of Novi Sad, Yugoslavia
gsilvia@uns.ns.ac.yu

Extended abstract

Abstract. We build two inverse limit lambda models which characterize completely sets of terms having similar computational behaviour. More precisely for each one 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. This is proved by using the finitary logical description of the models obtained by defining suitable intersection type assignment systems.

1 Introduction

The aim of this paper is to present two lambda models which completely characterize well-known computational properties of lambda terms. We consider nine computational properties of lambda terms and corresponding nine sets of lambda terms: the set of *normalizing*, *head normalizing*, *weak head normalizing* lambda terms, those corresponding to the *persistent* versions of these notions, and the sets of *closable*, *closable normalizing* and *closable head normalizing* lambda terms.

We build two *inverse lambda models* $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$, according to Scott [24], which completely characterize each of the mentioned sets of terms. More precisely for each one 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. First, we construct the sets $\mathcal{T}_\mathcal{D}$ and $\mathcal{T}_\mathcal{E}$ of types which are generated from atomic types corresponding to the elements of $\mathcal{D}_0$ and $\mathcal{E}_0$, by the *function type* constructor and the *intersection type* constructor. Then

⋆ Partially supported by EU within the FET - Global Computing initiative, project DART ST-2001-33477 , and by MURST Cofin'01 project COMETA. The funding bodies are not responsible for any use that might be made of the results presented here.

⋆⋆ 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).

we define the sets $\mathcal{F}_{\mathcal{D}}$ and $\mathcal{F}_{\mathcal{E}}$ of filters respectively on the sets $\mathcal{T}_{\mathcal{D}}$ and $\mathcal{T}_{\mathcal{E}}$. Following Scott [26], Coppo et al. [8], and Alessi [3], we will show that the sets $\mathcal{F}_{\mathcal{D}}$ (ordered by subset inclusion) and $\mathcal{F}_{\mathcal{E}}$ and the corresponding inverse models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are isomorphic as ω-algebraic cpos. This isomorphism falls in the general framework of *Stone dualities* (Johnstone [14]). This framework later received a categorically principled 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 lambda 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 (Alessi [3]). This gives the desired finitary logical descriptions of the models. Therefore an equivalent of the primary complete characterization can be stated: a term belongs to one of the nine sets mentioned if and only if it has a certain type (in a suitable context) in one of the obtained type assignment systems.

In order to prove one part of this property we apply the so called *reducibility method*. This method is a generally accepted way for proving the strong normalization property of various type systems (Tait [28], Tait [29], Girard [13], Krivine [16], [17], Mitchell [20]). The reducibility method is also used in Leivant [18] and Gallier [11] for characterizing strongly normalizing terms, normalizing terms, head normalizing terms, and weak head normalizing terms by their typeability in various intersection type systems. In Dezani et al. [10] the reducibility method is applied to characterizing both the mentioned sets of terms and their persistent versions.

In all these papers different properties are characterized by means of different type assignment systems: so the novelty of the present approach is that we characterize 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.

In the other direction of the proof the most intriguing part is the one concerning the persistently normalizing terms, which requires the characterization of these terms presented in Dezani et al. [10].

Lastly we remark that there are essentially *two* 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. [5], generalizes the one given by Scott for simple types (Scott [25]). 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 at the basis of our application of the reducibility method. The second semantics views types as *compact elements* of Plotkin's λ-structures (Plotkin [22]). According to this interpretation, the universal type denotes the least element, intersections denote joins of compact elements, and arrow types allow to internalize the space of continuous endomorphisms. This semantics allows us to obtain the isomorphisms between the models $\mathcal{D}_\infty$, $\mathcal{E}_\infty$ and the sets $\mathcal{F}_{\mathcal{D}}$, $\mathcal{F}_{\mathcal{E}}$ of filters of types.

The paper is organized as follows. In Section 2 the models $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ are built. The corresponding intersection type assignment systems are defined in Section 3. The main result is a complete characterization of computational behaviours of terms by their typeability in the corresponding type systems. This is stated in Section 4. For lack of space some proofs are omitted.

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) [9] and at the Types Workshop (TYPES 2002 24-28/04/2002, Nijmegen, The Netherlands).

2 The Models

We use standard notations for lambda terms and beta reductions.

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

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

We use $x, y, z, \ldots, x_1, \ldots$ for arbitrary term variables and $M, N, P, \ldots, M_1, \ldots$ for arbitrary terms. In writing terms we assume the standard conventions on parentheses and dots [6]. $\mathsf{FV}(M)$ denotes the set of free variables of a term M. By $M[x := N]$ we denote the term obtained by substituting the term N for all the free occurrences of the variable x in M, taking into account that free variables of N remain free in the term obtained.

The axiom of *β-reduction* is $(\lambda x.M)N \rightarrow_\beta M[x := N]$. A term of the form $(\lambda x.M)N$ is called a *β-redex*. The transitive reflexive closure of $\rightarrow_\beta$ is denoted by $\twoheadrightarrow_\beta$. A term is *a normal form* if it does not contain β-redexes.

We introduce now the computational behaviours of lambda terms we want to characterize.

Definition 2 (Normalization properties).

i) A term M has a normal form*, $M \in \mathcal{N}$, if M reduces to a normal form.*
ii) A term M has a head normal form*, $M \in \mathcal{HN}$, if M reduces to a term of the form $\lambda \vec{x}.y\vec{M}$ (where possibly y appears in $\vec{x}$).*
iii) A term M has a weak head normal form*, $M \in \mathcal{WN}$, if M 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 normalizing* terms have been introduced in Böhm and Dezani [7].

Definition 3 (Persistent normalization properties).

i) A term M is persistently normalizing*, $M \in \mathcal{PN}$, if $M\vec{N} \in \mathcal{N}$ for all terms $\vec{N}$ in $\mathcal{N}$.*
ii) A term M is persistently head normalizing*, $M \in \mathcal{PHN}$, if $M\vec{N} \in \mathcal{HN}$ for all terms $\vec{N}$.*
iii) A term M is persistently weak head normalizing*, $M \in \mathcal{PWN}$, if $M\vec{N} \in \mathcal{WN}$ for all terms $\vec{N}$.*

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

Definition 4 (Closability properties).

i) *A term M is* closable*, $M \in \mathcal{C}$, if M reduces to a closed term.*
ii) *A term M is* closable normalizing*, $M \in \mathcal{CN}$, if M reduces to a closed normal form.*
iii) *A term M is* closable head normalizing*, $M \in \mathcal{CHN}$, if M reduces to a closed head normal form.*

Example 1. Let $\mathbf{I} \equiv \lambda x.x,\ \ \Delta \equiv \lambda x.xx,\ \ \mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)),\ \ \mathbf{K} \equiv \lambda xy.x$.

– $\lambda x.x\Delta\Delta \in \mathcal{N}$, but $\lambda x.x\Delta\Delta \notin \mathcal{PWN}$ (hence $\lambda x.x\Delta\Delta \notin \mathcal{PHN}$), since $(\lambda x.x\Delta\Delta)\mathbf{I} \twoheadrightarrow_\beta \Delta\Delta \notin \mathcal{WN}$. Notice that $\lambda x.x\Delta\Delta \notin \mathcal{PN}$ since $\mathbf{I} \in \mathcal{N}$. Lastly $\lambda x.x\Delta\Delta \in \mathcal{CN}$.
– $\lambda x.y(\Delta\Delta) \in \mathcal{PHN}$, but $\lambda x.y(\Delta\Delta) \notin \mathcal{N}$.
– $\lambda x.x(\Delta\Delta) \in \mathcal{HN}$, but $\lambda x.x(\Delta\Delta) \notin \mathcal{N}$ and $\lambda x.x(\Delta\Delta) \notin \mathcal{PWN}$, since $(\lambda x.x(\Delta\Delta))\Delta \rightarrow_\beta \Delta(\Delta\Delta) \notin \mathcal{WN}$. Moreover $\lambda x.x(\Delta\Delta) \in \mathcal{CHN}$, but $\lambda x.x(\Delta\Delta) \notin \mathcal{CN}$.
– $\mathbf{YK} \in \mathcal{PWN}$, but $\mathbf{YK} \notin \mathcal{HN}$, hence $\mathbf{YK} \notin \mathcal{PHN}$.
– $\lambda x.\Delta\Delta \in \mathcal{WN}$, but $\lambda x.\Delta\Delta \notin \mathcal{HN}$ and $\lambda x.\Delta\Delta \notin \mathcal{PWN}$, since $(\lambda x.\Delta\Delta)M \twoheadrightarrow_\beta \Delta\Delta \notin \mathcal{WN}$. Moreover $\lambda x.\Delta\Delta \in \mathcal{C}$, but $\lambda x.\Delta\Delta \notin \mathcal{CHN}$, hence $\lambda x.\Delta\Delta \notin \mathcal{CN}$.

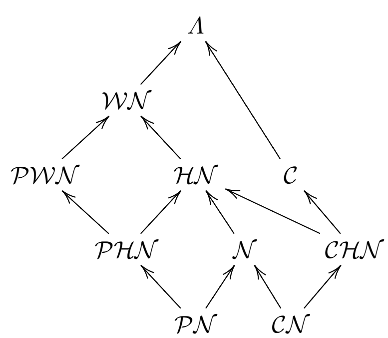

Fig. 1. Inclusions between sets of λ-terms

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

Proposition 1. *The following* strict *inclusions hold:*

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

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 normalizing term M is either an unsolvable term of order ∞ (as defined in Abramsky and Ong [2]), i.e. for all n there is N such that $M =_\beta \lambda x_1 \ldots x_n.N$, or it is a solvable term such that the head variable of its head normal form is free. In fact if M is an unsolvable term of a finite order, i.e. $M =_\beta \lambda x_1 \ldots x_n.N$ where N is unsolvable and it does not reduce to an abstraction, then $M\vec{N} \notin \mathcal{WN}$ where $\vec{N}$ are n arbitrary λ-terms. If $M =_\beta \lambda \vec{x}\, y\, \vec{z}.y\vec{N}$ we get $M\vec{X}(\Delta\Delta)\vec{Z} \twoheadrightarrow_\beta \Delta\Delta\vec{N'} \notin \mathcal{WN}$, where $\vec{X}$ has the same length as $\vec{x}$, $\vec{Z}$ has the same length of $\vec{z}$, Δ is defined in Example 1, and $\vec{N'} = \vec{N}[\vec{x} := \vec{X}, y := \Delta\Delta, \vec{z} := \vec{Z}]$.

The above discussion also shows that a persistently head normalizing 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 normalizing term to be a persistently normalizing term is that the head variable of its normal form is free. This condition is not sufficient, since for example $(\lambda x.y(xx))\Delta \to_\beta y(\Delta\Delta)$. 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:

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

The above discussion gives some inclusions between the current sets of terms, and Example 1 shows differences between them. The remaining inclusions easily follow by definition.

Our goal is to build two inverse limit lambda models (Scott [24]) 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, where as usual a step function $\mathsf{a} \Rightarrow \mathsf{b}$ is defined by

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

A weak head normalizing 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 $\bot \Rightarrow \bot$. So we can choose the representative of the step function $\bot \Rightarrow \bot$ as the element which corresponds to the elements of $\mathcal{WN}$. 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 normalizing term applied to any number of arbitrary terms gives a weak head normalizing term, i.e. it behaves at least as well as the step function $\underbrace{\bot \Rightarrow \ldots \Rightarrow \bot}_{n} \Rightarrow \bot$ for all values of 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 normalizing term when applied to a persistently head normalizing term reduces to a head normalizing term: in its turn a persistently head normalizing term applied to an arbitrary term gives a persistently head normalizing term. Therefore, if h and $\hat{\mathsf{h}}$ are two elements of $\mathcal{D}_0$ corresponding respectively to the sets $\mathcal{HN}$ and $\mathcal{PHN}$, they represent the step functions $\hat{\mathsf{h}} \Rightarrow \mathsf{h}$ and $\bot \Rightarrow \hat{\mathsf{h}}$.

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

Proposition 2. *The application of a normalizing term to a persistently normalizing term is in turn a normalizing term.*

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

Therefore if n and $\hat{\mathsf{n}}$ are two elements of $\mathcal{D}_0$ 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 gives a closable term. Then if c is the element representing $\mathcal{C}$ it behaves like the function $\mathsf{c} \Rightarrow \mathsf{c}$. The key observation here is that there are closable terms (like $\Delta\Delta$, where Δ is defined in Example 1) which are not weak head normalizing, and therefore we need to equate $\bot$ and $\bot \Rightarrow \bot$, i.e. we need to consider a solution of the domain equation $D = [D \to D]$. Moreover we do not have a join between c and $\hat{\mathsf{h}}$ (and hence $\hat{\mathsf{n}}$) since all persistently head normalizing terms are open. Therefore we consider a cpo $\mathcal{E}_0$ with elements $\mathsf{c}, \mathsf{n}, \hat{\mathsf{h}}, \hat{\mathsf{n}}$.

To sum up we define our models as follows.

Definition 5.

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

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

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

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

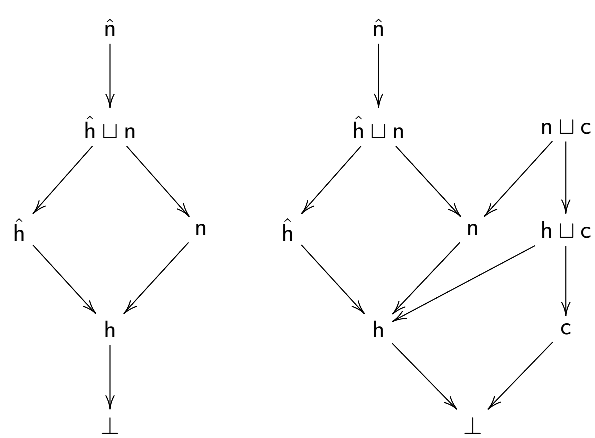

Fig. 2. The lattice $\mathcal{D}_0$ and the cpo $\mathcal{E}_0$

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 normalizing 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 1 (Main Theorem, Version I). *Let $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$ be the inverse limit models defined in Definition 5 and $\theta_{\hat{\mathsf{n}}}$ the environment defined by $\theta_{\hat{\mathsf{n}}}(x) = \hat{\mathsf{n}}$ for all $x \in \mathtt{var}$. Then:*

i) $M \in \mathcal{PN}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \hat{\mathsf{n}}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \hat{\mathsf{n}}$;
ii) $M \in \mathcal{N}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \mathsf{n}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{n}$;
iii) $M \in \mathcal{PHN}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \hat{\mathsf{h}}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \hat{\mathsf{h}}$;
iv) $M \in \mathcal{HN}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \mathsf{h}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{h}$;
v) $M \in \mathcal{PWN}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \bigsqcup_{n\in\mathbb{N}}(\underbrace{\bot \Rightarrow \ldots \Rightarrow \bot}_{n} \Rightarrow \bot)$;
vi) $M \in \mathcal{WN}$ *iff* $[\![M]\!]^{\mathcal{D}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{D}} \bot \Rightarrow \bot$;
vii) $M \in \mathcal{CN}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c} \sqcup \mathsf{n}$;
viii) $M \in \mathcal{CHN}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c} \sqcup \mathsf{h}$;
ix) $M \in \mathcal{C}$ *iff* $[\![M]\!]^{\mathcal{E}_\infty}_{\theta_{\hat{\mathsf{n}}}} \sqsupseteq^{\mathcal{E}} \mathsf{c}$.

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 3.

3 The Type Assignment System

Stone dualities allow to describe special classes of topological spaces by means of (possibly finitary) partial orders. Typically, these partial orders are given by the topology, a basis for it, or a subbasis for it. The seminal result is the duality between the categories of Stone spaces and of Boolean algebras (see Johnstone [14]). Other very important examples are the descriptions of *ω-algebraic complete lattices* as *intersection type theories* in Coppo et al. [8], *Scott domains* as *information systems* in Scott [26], 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 Martin-Löf [19].

As stated first in Coppo et al. [8] and proved in Alessi [3], we can describe an inverse limit model by taking:

- 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 iff the corresponding elements have a join;
- the preorder between types induced by reversing the order in the initial cpo and by encoding the initial projection, according to the correspondence:

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

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 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 projection the projection $\mathsf{i}_0^{\mathcal{E}}$ of Definition 5. We first define pretypes corresponding to the elements in $\widetilde{\mathcal{E}}_\infty$ and then types corresponding to the elements in $\mathcal{D}_\infty$ and $\mathcal{E}_\infty$.

Definition 6 (The set $\mathcal{PT}$ of pretypes). *The set $\mathcal{PT}$ of pretypes is defined as follows.*

$$\boxed{\mathcal{PT} ::= \nu \mid \hat{\nu} \mid \mu \mid \hat{\mu} \mid \gamma \mid \Omega \mid \mathcal{PT} \to \mathcal{PT} \mid \mathcal{PT} \cap \mathcal{PT}}$$

Pretypes will be denoted by $\phi, \phi_1, \ldots, \phi', \ldots$.

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

Definition 7 (The mapping $\mathtt{m}$). *The mapping $\mathtt{m} : \mathcal{PT} \to \widetilde{\mathcal{E}}_\infty$ is defined as follows.*

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

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

Definition 8 (The sets $\mathcal{T}_{\mathcal{D}}$ and $\mathcal{T}_{\mathcal{E}}$ of types).

i) A pretype ϕ is a $\mathcal{D}$-type, $\phi \in \mathcal{T}_{\mathcal{D}}$ iff $\mathtt{m}(\phi) \in \mathcal{D}_\infty$;
ii) A pretype ϕ is an $\mathcal{E}$-type, $\phi \in \mathcal{T}_{\mathcal{E}}$ iff $\mathtt{m}(\phi) \in \mathcal{E}_\infty$.

Types will be denoted by $\sigma, \tau, \ldots, \sigma_1, \ldots$. 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

$$(\sigma \to \tau \to \zeta) \cap \sigma \to \tau \to \zeta \equiv ((\sigma \to (\tau \to \zeta)) \cap \sigma) \to (\tau \to \zeta).$$

Moreover $\sigma^n \to \tau$ will be short for $\underbrace{\sigma \to \cdots \to \sigma}_{n} \to \tau \ (n \geq 0)$.

$(\hat{\nu}\nu)$	$\hat{\nu} \leq \nu$	$(\hat{\nu}\hat{\mu})$	$\hat{\nu} \leq \hat{\mu}$
$(\nu\mu)$	$\nu \leq \mu$	$(\hat{\mu}\mu)$	$\hat{\mu} \leq \mu$
(Ω)	$\sigma \leq \Omega$	(arint)	$(\sigma \to \tau) \cap (\sigma \to \zeta) \leq \sigma \to \tau \cap \zeta$
$(\hat{\nu}\to)$	$\hat{\nu} \sim (\Omega \to \hat{\mu}) \cap (\nu \to \hat{\nu})$	$(\nu \to)$	$\nu \sim (\hat{\mu} \to \mu) \cap (\hat{\nu} \to \nu)$
$(\hat{\mu}\to)$	$\hat{\mu} \sim \Omega \to \hat{\mu}$	$(\mu \to)$	$\mu \sim \hat{\mu} \to \mu$
(refl)	$\sigma \leq \sigma$	(mon)	$\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma \cap \tau \leq \sigma' \cap \tau'$
(idem)	$\sigma \leq \sigma \cap \sigma$	(trans)	$\sigma \leq \tau,\ \tau \leq \zeta \ \Rightarrow \sigma \leq \zeta$
(incl)	$\sigma \cap \tau \leq \sigma,\ \sigma \cap \tau \leq \tau$	(arco)	$\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma' \to \tau \leq \sigma \to \tau'$

where $\sigma \sim \tau$ is short for $\sigma \leq \tau$ and $\tau \leq \sigma$.

Fig. 3. Preorder axioms and rules

Figure 3 defines a preorder on pretypes. The first four axioms correspond to the partial order in $\mathcal{D}_0$ and $\mathcal{E}_0$. The axioms $(\hat{\nu} \to), (\nu \to), (\hat{\mu} \to), (\mu \to)$ encode the initial projection of the constants different from γ and Ω. The remaining axioms are standard properties of joins and step functions. We can now give the preorders on types.

Definition 9 (Preorders on $\mathcal{T}_\mathcal{D}$ and $\mathcal{T}_\mathcal{E}$).

i) *The relation $\leq_\mathcal{D}$ is defined on $\mathcal{T}_\mathcal{D}$ by the axioms and rules of Figure 3 plus the following axiom:*

$$(\Omega\rightarrow) \qquad \sigma\rightarrow\Omega \leq \Omega\rightarrow\Omega;$$

ii) *The relation $\leq_\mathcal{E}$ is defined on $\mathcal{T}_\mathcal{E}$ by the axioms and rules of Figure 3 plus the following axioms:*

$$(\rightarrow\Omega) \quad \Omega \leq \Omega\rightarrow\Omega \qquad (\gamma\rightarrow) \quad \gamma \sim \gamma\rightarrow\gamma.$$

The axioms $(\Omega\rightarrow)$ and $(\rightarrow\Omega)$ reflect the differences between the projections $\mathsf{i}_0^\mathcal{D}$ and $\mathsf{i}_0^\mathcal{E}$ on $\perp$. Notice that $(\rightarrow\Omega)$ and (Ω) imply $(\Omega\rightarrow)$. The axiom $(\gamma\rightarrow)$ encodes the initial projection of the constant γ.

Remark 1. The sets $\mathcal{T}_\mathcal{D}$ and $\mathcal{T}_\mathcal{E}$ are not closed under $\leq$, since for example by rule $(incl)$ $\hat{\nu}\cap\gamma \leq \hat{\nu}$, and $\hat{\nu}\in\mathcal{T}_\mathcal{D}$, $\hat{\nu}\in\mathcal{T}_\mathcal{E}$, while $\hat{\nu}\cap\gamma\notin\mathcal{T}_\mathcal{D}$, $\hat{\nu}\cap\gamma\notin\mathcal{T}_\mathcal{E}$. For this reason we take $\leq_\mathcal{D}$ (respectively $\leq_\mathcal{E}$) as the restriction of $\leq$ to $\mathcal{T}_\mathcal{D}$ (respectively $\mathcal{T}_\mathcal{E}$).

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

Definition 10 (The sets $\mathcal{F}_\mathcal{D}$ and $\mathcal{F}_\mathcal{E}$).

i) *A* filter *is a set $\Xi \subseteq \mathcal{PT}$ such that:*
  (a) $\Omega\in\Xi$;
  (b) *if $\sigma\leq\tau$ and $\sigma\in\Xi$, then $\tau\in\Xi$;*
  (c) *if $\sigma,\tau\in\Xi$, then $\sigma\cap\tau\in\Xi$;*
  where $\leq$ is the preorder defined in Figure 3;

ii) *if $\Xi\subseteq\mathcal{T}$, then $\uparrow\Xi$ denotes the filter generated by Ξ;*

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

iv) *$\mathcal{F}_\mathcal{D}$ denotes the set of filters Ξ such that*
  (a) *if $\sigma\in\Xi$, then $\sigma\in\mathcal{T}_\mathcal{D}$;*
  (b) *if $\sigma\leq_\mathcal{D}\tau$ and $\sigma\in\Xi$, then $\tau\in\Xi$;*

v) *$\mathcal{F}_\mathcal{E}$ denotes the set of filters Ξ such that*
  (a) *if $\sigma\in\Xi$, then $\sigma\in\mathcal{T}_\mathcal{E}$;*
  (b) *if $\sigma\leq_\mathcal{E}\tau$ and $\sigma\in\Xi$, then $\tau\in\Xi$.*

It is easy to verify that the set $\mathcal{F}_\mathcal{D}$, ordered by subset inclusion, is an ω-algebraic complete lattice, where $\uparrow\Omega$ is the bottom, and $\mathcal{T}$ is the top. Further, the set $\mathcal{F}_\mathcal{E}$, ordered by subset inclusion, is an ω-algebraic cpo, where $\uparrow\Omega$ is the bottom. For both domains the finite elements are exactly the principal filters.

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 ω-algebraic cpos. 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 models. Let $\nabla\in\{\mathcal{D},\mathcal{E}\}$. For lack of space the proof is omitted.

Lemma 1. *For all types $\sigma, \tau \in \mathcal{T}^{\nabla}$ we get:*

$$\mathtt{m}(\sigma) \sqsupseteq^{\nabla} \mathtt{m}(\tau) \textit{ iff } \sigma \leq_{\nabla} \tau.$$

Theorem 2 (Isomorphism).

i) The mapping $\mathtt{m}^* : \mathcal{F}_{\mathcal{D}} \to \mathcal{D}_{\infty}$ defined by

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

is a lattice isomorphism between $\mathcal{F}_{\mathcal{D}}$ and $\mathcal{D}_{\infty}$.

ii) The mapping $\mathtt{m}^* : \mathcal{F}_{\mathcal{E}} \to \mathcal{E}_{\infty}$ defined by

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

is a cpo isomorphism between $\mathcal{F}_{\mathcal{E}}$ and $\mathcal{E}_{\infty}$.

Proof. Clearly if $\Xi \subseteq \Xi'$ then $\bigsqcup_{\sigma \in \Xi} \mathtt{m}(\sigma) \sqsubseteq^{\nabla} \bigsqcup_{\tau \in \Xi'} \mathtt{m}(\tau)$.

Vice versa, if $\bigsqcup_{\sigma \in \Xi} \mathtt{m}(\sigma) \sqsubseteq^{\nabla} \bigsqcup_{\tau \in \Xi'} \mathtt{m}(\tau)$ then for all $\sigma \in \Xi$ there is $\tau \in \Xi'$ such that $\mathtt{m}(\sigma) \sqsubseteq^{\nabla} \mathtt{m}(\tau)$. By Lemma 1 this implies that for all $\sigma \in \Xi$ there is $\tau \in \Xi'$ such that $\sigma \geq^{\nabla} \tau$. Then being Ξ' a filter we get $\sigma \in \Xi'$ for all $\sigma \in \Xi$.

Notice that $\mathtt{m}^*(\uparrow \sigma) = \mathtt{m}(\sigma)$.

Due to the above isomorphism the interpretations of lambda 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 finitary logical descriptions of the models.

Let $\nabla \in \{\mathcal{D}, \mathcal{E}\}$. A ∇-*type assignment* is an expression of the form $M : \tau$, where $M \in \Lambda$ is the *subject* and $\tau \in \mathcal{T}_{\nabla}$ is the *predicate*. A ∇-*context* Γ is a (possibly infinite) set of ∇-type assignments of the shape $x : \sigma$ with different subjects (term variables).

In the following we will use both the contexts which assign to all variables the same type and the contexts which assign to all variables but one the same type.

Definition 11.
$\Gamma_\sigma = \{x : \sigma \mid x \in \mathtt{var}\}$ *and* $\Gamma_\sigma^{x:\tau} = \{x : \tau\} \cup \{y : \sigma \mid \ y \in \mathtt{var} \& y \neq x\}$.

Exploiting the intersection type constructor we can build a context out of two arbitrary contexts.

Definition 12.
$$\begin{aligned}\Gamma_1 \uplus \Gamma_2 = &\{(x{:}\tau) \mid (x{:}\tau) \in \Gamma_1 \& x \notin \Gamma_2\} \cup \\ &\{(x{:}\tau) \mid (x{:}\tau) \in \Gamma_2 \& x \notin \Gamma_1\} \cup \\ &\{(x{:}\tau_1 \cap \tau_2) \mid (x{:}\tau_1) \in \Gamma_1 \& (x{:}\tau_2) \in \Gamma_2\}\end{aligned}$$

Definition 13 (The type assignment systems). *The ∇-type assignment $M : \tau$ is derivable from the ∇-context Γ, notation $\Gamma \vdash_{\nabla} M : \tau$, if $\Gamma \vdash_{\nabla} M : \tau$ can be generated by the following axioms and rules.*

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} (ax) \qquad \frac{}{\Gamma \vdash M : \Omega} (\Omega)$$

$$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} (\to E) \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} (\to I)$$

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \cap \tau} (\cap I) \qquad \frac{\Gamma \vdash M : \sigma, \quad \sigma \leq_\nabla \tau}{\Gamma \vdash M : \tau} (\leq_\nabla)$$

It is easy to verify that the intersection elimination rule is derivable:

$$\frac{\Gamma \vdash M : \sigma \cap \tau}{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau} (\cap E)$$

and that the following rules are admissible:

$$\frac{\Gamma, x : \sigma \vdash M : \tau \quad \sigma' \leq_\nabla \sigma}{\Gamma, x : \sigma' \vdash M : \tau} (\leq_\nabla L) \qquad \frac{\Gamma \vdash M : \tau \quad x \notin \Gamma}{\Gamma, x : \sigma \vdash M : \tau} (weakening)$$

Example 2. Figure 4 gives some paradigmatic examples of deductions in our type systems. Notice the use of the intersection introduction and the 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 $\mathcal{E}$.

As usual we have a Generation Theorem for our type assignment system: the proof by induction on derivations follows the proof of the same property for the standard intersection type system (see e.g. [5]).

Theorem 3 (Generation Theorem).

1. *Assume $\sigma \not\sim^\nabla \Omega$. Then $\Gamma \vdash_\nabla x : \sigma$ iff $(x : \tau) \in \Gamma$ and $\tau \leq_\nabla \sigma$ for some $\tau \in \mathcal{T}^\nabla$.*
2. *$\Gamma \vdash_\nabla MN : \sigma$ iff $\Gamma \vdash_\nabla M : \tau \to \sigma$, and $\Gamma \vdash_\nabla N : \tau$ for some $\tau \in \mathcal{T}^\nabla$.*
3. *$\Gamma \vdash_\nabla \lambda x.M : \tau \to \sigma$ iff $\Gamma, x : \tau \vdash_\nabla M : \sigma$.*

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

Mappings $\theta : \mathtt{var} \to \mathcal{D}_\infty$ and $\theta : \mathtt{var} \to \mathcal{E}_\infty$ are called environments. The notation $\Gamma \rhd \theta$ means that $(x : \sigma) \in \Gamma$ implies $\mathtt{m}(\sigma) \sqsubseteq \theta(x)$.

Theorem 4 (Finitary logical descriptions).

i) For any lambda term M and environment $\theta : \mathtt{var} \to \mathcal{D}_\infty$,

$$[\![M]\!]^{\mathcal{D}_\infty}_\theta = \mathtt{m}^*(\{\tau \in \mathcal{T}_{\mathcal{D}} \mid \exists\Gamma.\ \Gamma \rhd \theta \ \&\ \Gamma \vdash_{\mathcal{D}} M : \tau\});$$

ii) For any lambda term M and environment $\theta : \mathtt{var} \to \mathcal{E}_\infty$,

$$[\![M]\!]^{\mathcal{E}_\infty}_\theta = \mathtt{m}^*(\{\tau \in \mathcal{T}_{\mathcal{E}} \mid \exists\Gamma.\ \Gamma \rhd \theta \ \&\ \Gamma \vdash_{\mathcal{E}} M : \tau\}).$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash y : \hat{\nu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash y : \Omega \to \hat{\mu}}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash x : \Omega}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash yx : \hat{\mu}}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.yx : \Omega \to \hat{\mu}}\,(\to I) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\nu} \vdash y : \hat{\nu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\nu} \vdash y : \nu \to \hat{\nu}}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\nu} \vdash x : \nu}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\nu} \vdash yx : \hat{\nu}}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.yx : \nu \to \hat{\nu}}\,(\to I)}{\dfrac{\Gamma_{\hat{\nu}} \vdash \lambda x.yx : (\Omega \to \hat{\mu}) \cap (\nu \to \hat{\nu})}{\Gamma_{\hat{\nu}} \vdash \lambda x.yx : \hat{\nu}}\,(\leq)}\,(\cap I)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x : \hat{\mu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x : \hat{\mu} \to \mu}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x : \hat{\mu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash xx : \mu}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \hat{\mu} \to \mu}\,(\to I) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}} \vdash x : \hat{\nu}}\,(ax)}{\Gamma_{\hat{\nu}} \vdash x : \hat{\nu} \to \nu}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}} \vdash x : \hat{\nu}}\,(ax)}{\Gamma_{\hat{\nu}} \vdash xx : \nu}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \hat{\nu} \to \nu}\,(\to I)}{\dfrac{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : (\hat{\mu} \to \mu) \cap (\hat{\nu} \to \nu)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \nu}\,(\leq)}\,(\cap I)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash y : \hat{\nu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash y : \Omega \to \hat{\mu}}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash \Delta\Delta : \Omega}\,(\Omega)}{\Gamma_{\hat{\nu}}^{x:\Omega} \vdash y(\Delta\Delta) : \hat{\mu}}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.y(\Delta\Delta) : \Omega \to \hat{\mu}}\,(\to I)}{\Gamma_{\hat{\nu}} \vdash \lambda x.y(\Delta\Delta) : \hat{\mu}}\,(\leq) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x : \hat{\mu}}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x : \Omega \to \mu}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash \Delta\Delta : \Omega}\,(\Omega)}{\Gamma_{\hat{\nu}}^{x:\hat{\mu}} \vdash x(\Delta\Delta) : \mu}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.x(\Delta\Delta) : \hat{\mu} \to \mu}\,(\to I)}{\Gamma_{\hat{\nu}} \vdash \lambda x.x(\Delta\Delta) : \mu}\,(\leq)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma \to \gamma}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash xx : \gamma}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \gamma \to \gamma}\,(\to I) \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma \to \gamma}\,(\leq) \quad \dfrac{}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash x : \gamma}\,(ax)}{\Gamma_{\hat{\nu}}^{x:\gamma} \vdash xx : \gamma}\,(\to E)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \gamma \to \gamma}\,(\to I)}{\Gamma_{\hat{\nu}} \vdash \lambda x.xx : \gamma}\,(\leq)}{\Gamma_{\hat{\nu}} \vdash (\lambda x.xx)(\lambda x.xx) : \gamma}\,(\to E)$$

Fig. 4. Type derivations

Proof. Clearly it suffices to prove that

$$\mathtt{m}(\tau) \sqsubseteq^{\nabla} [\![M]\!]^{\nabla\infty}_{\theta} \text{ iff there is } \Gamma \rhd \theta \text{ such that } \Gamma \vdash_{\nabla} M : \tau.$$

The proof is by structural induction on M. The case $\tau \sim^{\nabla} \Omega$ is trivial.

If M is a variable then it follows immediately from the definition of $\Gamma \rhd \theta$.

If $M \equiv \lambda x.N$ let $\tau \sim^{\nabla} \bigcap_{i \in I}(\sigma_i \to \zeta_i)$. By definition of $[\![]\!]^{\nabla\infty}_{\theta}$: $\bigsqcup_{i \in I}(\mathtt{m}(\sigma_i) \Rightarrow \mathtt{m}(\zeta_i)) \sqsubseteq^{\nabla} [\![M]\!]^{\nabla\infty}_{\theta}$ iff $\mathtt{m}(\zeta_i) \sqsubseteq^{\nabla} [\![N]\!]^{\nabla\infty}_{\theta[x:=\mathtt{m}(\sigma_i)]}$ for all $i \in I$. By induction $\mathtt{m}(\zeta_i) \sqsubseteq^{\nabla} [\![N]\!]^{\nabla\infty}_{\theta[x:=\mathtt{m}(\sigma_i)]}$ iff there is $\Gamma' \rhd \theta[x := \mathtt{m}(\sigma_i)]$ such that $\Gamma' \vdash_{\nabla} N : \zeta_i$. By definition of $\rhd$ and Lemma 1 $\Gamma' \rhd \theta[x := \mathtt{m}(\sigma_i)]$ implies $\Gamma' = \Gamma, x : \sigma'_i$ for some $\sigma'_i \geq_{\nabla} \sigma_i$. We get by the Generation Theorem (Theorem 3) $\Gamma' \vdash_{\nabla} N : \zeta_i$ iff $\Gamma \vdash_{\nabla} M : \sigma_i \to \zeta_i$. Lastly $\Gamma \vdash_{\nabla} M : \sigma_i \to \zeta_i$ for all $i \in I$ iff $\Gamma \vdash_{\nabla} M : \tau$.

If $M \equiv NP$ by definition of $[\![]\!]^{\nabla\infty}_{\theta}$: $\mathtt{m}(\tau) \sqsubseteq^{\nabla} [\![M]\!]^{\nabla\infty}_{\theta}$ iff there is σ such that $\mathtt{m}(\sigma) \Rightarrow \mathtt{m}(\tau) \sqsubseteq^{\nabla} [\![N]\!]^{\nabla\infty}_{\theta}$ and $\mathtt{m}(\sigma) \sqsubseteq^{\nabla} [\![P]\!]^{\nabla\infty}_{\theta}$. By induction $\mathtt{m}(\sigma) \Rightarrow \mathtt{m}(\tau) \sqsubseteq^{\nabla} [\![N]\!]^{\nabla\infty}_{\theta}$ iff there is $\Gamma' \rhd \theta$ such that $\Gamma' \vdash_{\nabla} N : \sigma \to \tau$. Similarly $\mathtt{m}(\sigma) \sqsubseteq^{\nabla} [\![P]\!]^{\nabla\infty}_{\theta}$ iff there is $\Gamma'' \rhd \theta$ such that $\Gamma'' \vdash_{\nabla} P : \sigma$. Taking $\Gamma = \Gamma' \uplus \Gamma''$ we have $\Gamma \vdash_{\nabla} N : \sigma \to \tau$ and $\Gamma \vdash_{\nabla} P : \sigma$. By the Generation Theorem (Theorem 3) $\Gamma \vdash_{\nabla} N : \sigma \to \tau$ and $\Gamma \vdash_{\nabla} P : \sigma$ for some σ iff $\Gamma \vdash_{\nabla} M : \tau$.

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

Corollary 1. *If $\Gamma \vdash_{\nabla} M : \tau$ and $M =_{\beta} N$, then $\Gamma \vdash_{\nabla} N : \tau$.*

4 The Main Result

Theorem 4 allows us to rephrase the main theorem of previous section, Theorem 1, as follows:

Theorem 5 (Main Theorem, Version II).

i) $M \in \mathcal{PN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \hat{\nu}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \hat{\nu}$;
ii) $M \in \mathcal{N}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \nu$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \nu$;
iii) $M \in \mathcal{PHN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \hat{\mu}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \hat{\mu}$;
iv) $M \in \mathcal{HN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \mu$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \mu$;
v) $M \in \mathcal{PWN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \Omega^n \to \Omega$ *for all* $n \in \mathbb{N}$;
vi) $M \in \mathcal{WN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \Omega \to \Omega$;
vii) $M \in \mathcal{CN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \gamma \cap \nu$;
viii) $M \in \mathcal{CHN}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \gamma \cap \mu$;
ix) $M \in \mathcal{C}$ *iff* $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \gamma$.

The proofs of the *if* parts of this Theorem are mainly straightforward induction and case split, with the exception of the case of persistently normalizing terms. This case needs the characterization of the set $\mathcal{PN}$ given in Böhm and Dezani [10] and it is omitted. The proofs of the *only if* parts require the set-theoretic semantics of intersection types using saturated sets.

Proof of Theorem 5 (ix)-(ii)(⇒)
In this proof we will use the characterizations of $\mathcal{PWN}$ and $\mathcal{PHN}$ given in the proof of Proposition 1.

- (*ix*) We will show $\Gamma_\gamma \vdash_{\mathcal{E}} M : \gamma$ for all M by structural induction on M. If M is a variable it is trivial. If $M \equiv NP$, then by induction $\Gamma_\gamma \vdash_{\mathcal{E}} N : \gamma$ and $\Gamma_\gamma \vdash_{\mathcal{E}} P : \gamma$. By rule $(\leq_{\mathcal{E}})$ we get $\Gamma_\gamma \vdash_{\mathcal{E}} N : \gamma \to \gamma$ and therefore using $(\to E)$ we conclude $\Gamma_\gamma \vdash_{\mathcal{E}} M : \gamma$. If $M \equiv \lambda x.N$, then by induction $\Gamma_\gamma \vdash_{\mathcal{E}} N : \gamma$. Then using $(\to I)$ we deduce $\Gamma_\gamma \vdash_{\mathcal{E}} M : \gamma \to \gamma$, and we conclude by $(\leq_{\mathcal{E}})$ $\Gamma_\gamma \vdash_{\mathcal{E}} M : \gamma$. We can conclude $\vdash_{\mathcal{E}} M : \gamma$ for all closed M, and by rule $(weakening)$ $\Gamma_{\hat{\nu}} \vdash_{\mathcal{E}} M : \gamma$.
- (*vi*) By Corollary 1 it suffices to consider M in weak head normal form. If $M \equiv \lambda x.N$, then we get $\Gamma_{\hat{\nu}}^{x:\Omega} \vdash_{\mathcal{D}} N : \Omega$ by (Ω) and $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \Omega \to \Omega$ by rule $(\to I)$. If $M \equiv x\vec{N}$, where m is the length of $\vec{N}$, being $\hat{\nu} \leq_{\mathcal{D}} \Omega^{m+1} \to \Omega$, we derive $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} M : \Omega \to \Omega$ using (Ω), $(\leq_{\mathcal{D}})$ and $(\to E)$.
- (*v*) If M is an unsolvable term of order ∞, i.e. for all n, there is N such that $M =_\beta \lambda x_1 \ldots x_n.N$, we can derive $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} \lambda x_1 \ldots x_n.N : \Omega^n \to \Omega$ by (Ω) and rule $(\to I)$. If M is a solvable term such that the head variable of its head normal form is free, i.e. $M =_\beta \lambda \vec{x}.y\vec{N}$, being $\hat{\nu} \leq_{\mathcal{D}} \Omega^{m+l} \to \Omega$ for all l, we can derive $\Gamma_{\hat{\nu}} \vdash_{\mathcal{D}} \lambda \vec{x}.y\vec{N} : \Omega^{n+l} \to \Omega$, where m is the length of $\vec{N}$ and n is the length of $\vec{x}$.
- (*iv*) Again by Corollary 1 it suffices to consider M in head normal form. Let $M \equiv \lambda \vec{y}.x\vec{N}$ where $\vec{y}$ has length n and $\vec{N}$ has length m. We have $\Gamma_{\hat{\mu}} \vdash_{\nabla} x\vec{N} : \mu$ by rules (Ω), $(\leq_\nabla)$ and $(\to E)$ being $\hat{\mu} \leq_\nabla \Omega^m \to \mu$. By $(\to I)$ this implies $\Gamma_{\hat{\mu}} \vdash_\nabla M : \hat{\mu}^n \to \mu$. We conclude $\Gamma_{\hat{\nu}} \vdash_\nabla M : \mu$ using $(\leq_\nabla)$ and $(\leq_\nabla L)$.
- (*viii*) follows from (*iv*) and (*ix*) being $\mathcal{CHN} = \mathcal{C} \cap \mathcal{HN}$.
- (*iii*) The head variable of the head normal form of M must be free. We can type a term of the shape $\lambda \vec{x}.y\vec{N}$, where $y \notin \vec{x}$ as follows: $\Gamma_{\hat{\nu}} \vdash_\nabla \lambda \vec{x}.y\vec{N} : \Omega^n \to \hat{\mu}$, since $\hat{\nu} \leq_\nabla \Omega^m \to \hat{\mu}$, where m is the length of $\vec{N}$ and n is the length of $\vec{x}$. We conclude $\Gamma_{\hat{\nu}} \vdash_\nabla M : \hat{\mu}$ using $(\leq_\nabla)$.
- (*ii*) By (*iv*) we get $\Gamma_{\hat{\nu}} \vdash_\nabla M : \mu$. So we only need to prove $\Gamma_{\hat{\nu}} \vdash_\nabla M : \hat{\nu} \to \nu$. The proof is by induction on the normal form M. If M is a variable it is trivial since $\hat{\nu} \leq_\nabla \hat{\nu} \to \nu$. If $M \equiv x\vec{N}$ then by induction $\Gamma_{\hat{\nu}} \vdash_\nabla N : \nu$ for all $N \in \vec{N}$ and we get $\Gamma_{\hat{\nu}} \vdash_\nabla M : \hat{\nu} \to \nu$ since $\hat{\nu} \leq_\nabla \nu^m \to \hat{\nu} \to \nu$, where m is the length of $\vec{N}$. If $M \equiv \lambda x.N$ then by induction $\Gamma_{\hat{\nu}} \vdash_\nabla N : \nu$ and this gives $\Gamma_{\hat{\nu}} \vdash_\nabla M : \hat{\nu} \to \nu$ by rule $(\to I)$.
- (*vii*) follows from (*ii*) and (*ix*) being $\mathcal{CN} = \mathcal{C} \cap \mathcal{N}$.

In order to prove the (⇐)-part of our main statement (Theorem 5) 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 [28] for proving the strong normalization property of simply typed lambda calculus. Further it was developed in Tait [29] and Girard [13] for proving the strong normalization property of polymorphic lambda calculus.

In Pottinger [23], van Bakel [30], Krivine [16], [17], Ghilezan [12], Amadio and Curien [4], the reducibility method is applied in order to characterize all and only the strongly normalizing lambda terms in lambda calculus with intersection types. The reducibility method is also used for characterizing some special classes of lambda terms such as strongly normalizing terms, normalizing terms, head normalizing terms, and weak head normalizing terms. They are characterized by their typeability in various intersection type assignment systems in Leivant [18] and Gallier [11], whereas both the mentioned terms as well as their persistent versions are characterized in Dezani et al. [10]. Furthermore, this method was applied for the proof of the Church-Rosser property (confluence) of the simply typed lambda calculus in Statman [27], Koletsos [15], and Mitchell [20], [21].

We will adapt the reducibility method, by requiring that the terms typable with the key types listed in Theorem 5 belong to the corresponding sets.

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

We first define the *interpretations of types* in $\mathcal{T}_\mathcal{D}$ and in $\mathcal{T}_\mathcal{E}$: the only difference between the two interpretations concerns the arrow constructor.

Definition 14.

i) The map $[\![-]\!]^\mathcal{D} : \mathcal{T}_\mathcal{D} \to 2^\Lambda$ *is defined by:*
$[\![\nu]\!]^\mathcal{D} = \mathcal{N}$, $[\![\hat{\nu}]\!]^\mathcal{D} = \mathcal{PN}$, $[\![\mu]\!]^\mathcal{D} = \mathcal{HN}$, $[\![\hat{\mu}]\!]^\mathcal{D} = \mathcal{PHN}$, $[\![\Omega]\!]^\mathcal{D} = \Lambda$;
$[\![\sigma \cap \tau]\!]^\mathcal{D} = [\![\sigma]\!]^\mathcal{D} \cap [\![\tau]\!]^\mathcal{D}$;
$[\![\sigma \to \tau]\!]^\mathcal{D} = [\![\sigma]\!]^\mathcal{D} \overset{\mathcal{D}}{\longmapsto} [\![\tau]\!]^\mathcal{D} = \{M \in \mathcal{WN} \mid \forall N \in [\![\sigma]\!]^\mathcal{D} \quad MN \in [\![\tau]\!]^\mathcal{D}\}$.

ii) The map $[\![-]\!]^\mathcal{E} : \mathcal{T}_\mathcal{E} \to 2^\Lambda$ *is defined by:*
$[\![\nu]\!]^\mathcal{E} = \mathcal{N}$, $[\![\hat{\nu}]\!]^\mathcal{E} = \mathcal{PN}$, $[\![\mu]\!]^\mathcal{E} = \mathcal{HN}$ $[\![\hat{\mu}]\!]^\mathcal{E} = \mathcal{PHN}$, $[\![\gamma]\!]^\mathcal{E} = \mathcal{C}$,
$[\![\Omega]\!]^\mathcal{E} = \Lambda$;
$[\![\sigma \cap \tau]\!]^\mathcal{E} = [\![\sigma]\!]^\mathcal{E} \cap [\![\tau]\!]^\mathcal{E}$;
$[\![\sigma \to \tau]\!]^\mathcal{E} = [\![\sigma]\!]^\mathcal{E} \overset{\mathcal{E}}{\longmapsto} [\![\tau]\!]^\mathcal{E} = \{M \in \Lambda \mid \forall N \in [\![\sigma]\!]^\mathcal{E} \quad MN \in [\![\tau]\!]^\mathcal{E}\}$.

Notice that

$$\begin{aligned}&[\![\Omega \to \Omega]\!]^\mathcal{D} = \{M \in \mathcal{WN} \mid \forall N \in [\![\Omega]\!]^\mathcal{D}\ MN \in [\![\Omega]\!]^\mathcal{D}\}\\ &= \{M \in \mathcal{WN} \mid \forall N \in \Lambda\ MN \in \Lambda\} = \mathcal{WN},\end{aligned}$$

and by Definition 3

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

The following definition of *saturated set* is standard, see Krivine [16], [17].

Definition 15. *A set* $X \subseteq \Lambda$ *is* saturated*, notation SAT*(X)*, if*

$$(\forall M, N \in \Lambda)\,(\forall \vec{M} \in \Lambda)\ M[x := N]\vec{M} \in X\ \Rightarrow\ (\lambda x.M)N\vec{M} \in X.$$

Obviously, each one 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 2. $(\forall \tau \in \mathcal{T}_{\mathcal{D}})$*SAT*$(\llbracket \tau \rrbracket^{\mathcal{D}})$ *and* $(\forall \tau \in \mathcal{T}_{\mathcal{E}})$*SAT*$(\llbracket \tau \rrbracket^{\mathcal{E}})$.

Proof. The proof is by structural induction on types. The only interesting case is that of arrow types. Let $M, N, \vec{P} \in \Lambda$. Suppose $M[x := N]\vec{P} \in \llbracket \sigma \to \tau \rrbracket^{\mathcal{D}}$. Let $Q \in \llbracket \sigma \rrbracket^{\mathcal{D}}$ be arbitrary. Therefore by Definition 14(i) $M[x := N]\vec{P}Q \in \llbracket \tau \rrbracket^{\mathcal{D}}$. Then by the induction hypothesis $(\lambda x.M)N\vec{P}Q \in \llbracket \tau \rrbracket^{\mathcal{D}}$. Moreover by Definition 14(i) we get $M[x := N]\vec{P} \in \mathcal{WN}$, and this implies $(\lambda x.M)N\vec{P} \in \mathcal{WN}$. Since Q was arbitrary, according to Definition 14(i) we get $(\lambda x.M)N\vec{P} \in \llbracket \sigma \to \tau \rrbracket^{\mathcal{D}}$. Similarly one can show that $M[x := N]\vec{P} \in \llbracket \sigma \to \tau \rrbracket^{\mathcal{E}}$ implies $(\lambda x.M)N\vec{P} \in \llbracket \sigma \to \tau \rrbracket^{\mathcal{E}}$.

We can get a simplification of Lemma 2.

Corollary 2. $(\forall \tau \in \mathcal{T}^{\nabla})$ $(\forall N \in \Lambda)$ $M[x := N] \in \llbracket \tau \rrbracket^{\nabla} \Rightarrow (\lambda x.M)N \in \llbracket \tau \rrbracket^{\nabla}$.

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

Lemma 3. *If* $\sigma \leq_{\nabla} \tau$*, then* $\llbracket \sigma \rrbracket^{\nabla} \subseteq \llbracket \tau \rrbracket^{\nabla}$.

Proof. By induction on the length of the derivation of $\sigma \leq_{\nabla} \tau$. Proposition 1 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 2 taking into account that $\mathcal{N} \subseteq \mathcal{PHN} \to \mathcal{HN}$ since $\mathcal{HN} = \mathcal{PHN} \to \mathcal{HN}$.

Let us further define the ∇-*valuations of terms* $\llbracket - \rrbracket^{\nabla}_{\rho} : \Lambda \to \Lambda$ and the *semantic satisfiability relations* $\models_{\nabla}$ which connect the type interpretations and the term valuations as follows.

Definition 16. *Let* $\llbracket - \rrbracket^{\nabla} : \mathcal{T}_{\nabla} \to 2^{\Lambda}$*,* $\nabla \in \{\mathcal{D}, \mathcal{E}\}$*, be the defined type interpretation and let* $\rho : \mathtt{var} \to \Lambda$ *be a valuation of term variables in* Λ*. Then*

i) $\llbracket - \rrbracket^{\nabla}_{\rho} : \Lambda \to \Lambda$ *is defined by*
$\llbracket M \rrbracket^{\nabla}_{\rho} = M[x_1 := \rho(x_1), \dots, x_n := \rho(x_n)]$*, where* $\mathsf{FV}(M) = \{x_1, \dots, x_n\}$*;*
ii) $\rho \models_{\nabla} M : \tau \quad$ *iff* $\quad \llbracket M \rrbracket^{\nabla}_{\rho} \in \llbracket \tau \rrbracket^{\nabla}$*;*
iii) $\rho \models_{\nabla} \Gamma \quad$ *iff* $\quad (\forall (x : \sigma) \in \Gamma) \quad \rho \models_{\nabla} x : \sigma$*;*
iv) $\Gamma \models_{\nabla} M : \tau \quad$ *iff* $\quad (\forall \rho \models_{\nabla} \Gamma) \quad \rho \models_{\nabla} M : \tau$.

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

Theorem 6 (Soundness).

$$\Gamma \vdash_{\mathcal{D}} M : \tau \Rightarrow \Gamma \models_{\mathcal{D}} M : \tau \qquad\qquad \Gamma \vdash_{\mathcal{E}} M : \tau \Rightarrow \Gamma \models_{\mathcal{E}} M : \tau.$$

Proof. By induction on the derivation of $\Gamma \vdash_\nabla M : \tau$.

Case 1. The last step applied is (ax), i.e. $\Gamma, x : \tau \vdash_\nabla x : \tau$. Then $\Gamma, x : \tau \models_\nabla x : \tau$ by Definition 16(*iii*).

Case 2. The last step applied is $(\rightarrow E)$, i.e. $\Gamma \vdash_\nabla M : \sigma \rightarrow \tau, \Gamma \vdash_\nabla N : \sigma \Rightarrow \Gamma \vdash_\nabla MN : \tau$. Then by the induction hypothesis $\Gamma \models_\nabla M : \sigma \rightarrow \tau$ and $\Gamma \models_\nabla N : \sigma$. Let $\rho \models_\nabla \Gamma$, then $[\![M]\!]^\nabla_\rho \in [\![\sigma \rightarrow \tau]\!]^\nabla = [\![\sigma]\!]^\nabla \overset{\nabla}{\longmapsto} [\![\tau]\!]^\nabla$ and $[\![N]\!]^\nabla_\rho \in [\![\sigma]\!]^\nabla$. Therefore $[\![MN]\!]^\nabla_\rho \equiv [\![M]\!]^\nabla_\rho [\![N]\!]^\nabla_\rho \in [\![\tau]\!]^\nabla$.

Case 3. The last step applied is $(\rightarrow I)$, i.e. $\Gamma, x : \sigma \vdash_\nabla M : \tau \Rightarrow \Gamma \vdash_\nabla \lambda x.M : \sigma \rightarrow \tau$. By the induction hypothesis $\Gamma, x : \sigma \models_\nabla M : \tau$. Let $\rho \models_\nabla \Gamma$ and let $N \in [\![\sigma]\!]^\nabla$. We define $\rho[x := N](x) = N, \rho[x := N](y) = \rho(y)$ for $x \neq y$. Then $\rho[x := N] \models_\nabla \Gamma$, since $x \notin \Gamma$, and $\rho[x := N] \models_\nabla x : \sigma$, since $N \in [\![\sigma]\!]^\nabla$. Therefore $\rho[x := N] \models_\nabla M : \tau$, i.e. $[\![M]\!]^\nabla_{\rho[x := N]} \in [\![\tau]\!]^\nabla$, which means by Definition 16(*i*) that $M[\vec{y} := \rho(\vec{y})][x := N] \in [\![\tau]\!]^\nabla$, where $\vec{y} = \mathsf{FV}(M) \setminus \{x\}$. By Corollary 2 we have $(\lambda x.M[\vec{y} := \rho(\vec{y})])N \in [\![\tau]\!]^\nabla$. Then $[\![\lambda x.M]\!]^\nabla_\rho N \in [\![\tau]\!]^\nabla$ since $x \notin \mathsf{FV}(\lambda x.M)$. Notice that $[\![\lambda x.M]\!]^\mathcal{D}_\rho \in \mathcal{WN}$. Therefore $[\![\lambda x.M]\!]^\mathcal{D}_\rho \in [\![\sigma]\!]^\mathcal{D} \overset{\mathcal{D}}{\longmapsto} [\![\tau]\!]^\mathcal{D} = [\![\sigma \rightarrow \tau]\!]^\mathcal{D}$ since $N \in [\![\sigma]\!]^\mathcal{D}$ was arbitrary. Similarly $[\![\lambda x.M]\!]^\mathcal{E}_\rho \in [\![\sigma \rightarrow \tau]\!]^\mathcal{E}$.

Case 4. The last step applied is $(\cap I)$, i.e. $\Gamma \vdash_\nabla M : \sigma,\ \Gamma \vdash_\nabla M : \tau \Rightarrow \Gamma \vdash_\nabla M : \sigma \cap \tau$. Then by the induction hypothesis $\Gamma \models_\nabla M : \sigma$ and $\Gamma \models_\nabla M : \tau$. Let $\rho \models_\nabla \Gamma$, then $[\![M]\!]^\nabla_\rho \in [\![\sigma]\!]^\nabla$ and $[\![M]\!]^\nabla_\rho \in [\![\tau]\!]^\nabla$. Therefore $[\![M]\!]^\nabla_\rho \in [\![\sigma \cap \tau]\!]^\nabla$, i.e. $\Gamma \models_\nabla M : \sigma \cap \tau$.

Case 5. The last step applied is $(\leq_\nabla)$, i.e. $\Gamma \vdash_\nabla M : \sigma,\ \sigma \leq_\nabla \tau \Rightarrow \Gamma \vdash_\nabla M : \tau$. By the induction hypothesis $\Gamma \models_\nabla M : \sigma$. Let $\rho \models_\nabla \Gamma$, then $[\![M]\!]^\nabla_\rho \in [\![\sigma]\!]^\nabla$. According to Lemma 3 $[\![\sigma]\!]^\nabla \subseteq [\![\tau]\!]^\nabla$ so it follows that $[\![M]\!]^\nabla_\rho \in [\![\tau]\!]^\nabla$, i.e. $\Gamma \models_\nabla M : \tau$.

We conclude this section with the proof of the *only if part* of Main Theorem, Version II.

Proof of Theorem 5 ($\Leftarrow$).
The proofs of all parts are similar, so we only consider part (v). Let $\Gamma_{\hat{\nu}} \vdash M : \Omega^n \rightarrow \Omega$, for all n: by soundness (Theorem 6) we have that if $\rho \models_\mathcal{D} \Gamma_{\hat{\nu}}$, then $[\![M]\!]^\mathcal{D}_\rho \in [\![\Omega^n \rightarrow \Omega]\!]^\mathcal{D}$, for all n. We can take $\rho_1(x) = x$, being $\rho_1 \models_\mathcal{D} \Gamma_{\hat{\nu}}$, because all variables belong to $\mathcal{PN}$. Obviously, $\rho_1(M) = M$ for every lambda term M. Therefore we get that $M \in [\![\Omega^n \rightarrow \Omega]\!]^\mathcal{D}$, for all n: hence, $M \in \mathcal{PWN}$ since $\bigcap_{n \in \mathbb{N}} [\![\Omega^n \rightarrow \Omega]\!]^\mathcal{D} = \mathcal{PWN}$ by Definition 14.

5 Discussion

Two natural questions, at least, lurk behind this paper: "can we characterize in some significant way the class of evaluation properties which we can characterize using lambda models?" and "is there a method for going from a logical specification of a property to the appropriate lambda model?".

Regarding the first question, obviously the sets of terms having a given property have to be closed, at least, under β-conversion. But clearly this is not the whole story. Probably the answer to this question is linked to some very important open problems in the theory of the denotational semantics of untyped λ-calculus, like the existence of a denotational model whose theory is precisely $\lambda\beta$. As far as the latter question is concerned, we really have no idea. It seems that we are still missing something in our understanding of lambda models.

The main contribution of the present paper is to show that two models can characterize many different sets of terms. On the one hand it seems that we cannot find elements representing weak head normalizability 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 normalization property with its persistent version. It is not clear if these properties can be characterized separately, i.e. if one can build models in which only one of these properties is characterized.

Acknowledgements

We are grateful to the referees for careful reading and useful suggestions.

References

1. Samson Abramsky. Domain theory in logical form. *Ann. Pure Appl. Logic*, 51(1-2):1–77, 1991.
2. Samson Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Inform. and Comput.*, 105(2):159–267, 1993.
3. Fabio Alessi. *Strutture di tipi, teoria dei domini e modelli del lambda calcolo*. PhD thesis, Torino University, 1991.
4. Roberto M. Amadio and Pierre-Louis Curien. *Domains and Lambda-Calculi*. Cambridge University Press, Cambridge, 1998.
5. Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *J. Symbolic Logic*, 48(4):931–940 (1984), 1983.
6. Henk P. Barendregt. *The Lambda Calculus: its Syntax and Semantics*. North-Holland, Amsterdam, revised edition, 1984.
7. Corrado Böhm and Mariangiola 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.
8. Mario Coppo, Mariangiola Dezani-Ciancaglini, Furio Honsell, and Giuseppe 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.
9. Mariangiola Dezani-Ciancaglini and Silvia Ghilezan. A lambda model characterizing computational behaviours of terms. In Y.Toyama, editor, *International Workshop on Rewriting in Proof and Computation, RPC'01*, pages 100–119, 2001.
10. Mariangiola Dezani-Ciancaglini, Furio Honsell, and Yoko Motohama. Compositional characterization of λ-terms using intersection types. In *Mathematical Foundations of Computer Science 2000*, volume 1893 of *Lecture Notes in Computer Science*, pages 304–313. Springer-Verlag, 2000.
11. Jean Gallier. Typing untyped λ-terms, or reducibility strikes again! *Ann. Pure Appl. Logic*, 91:231–270, 1998.

12. Silvia Ghilezan. Strong normalization and typability with intersection types. *Notre Dame J. Formal Logic*, 37(1):44–52, 1996.
13. Jean-Yves 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. North-Holland, 1971.
14. Peter T. Johnstone. *Stone Spaces*. Cambridge University Press, Cambridge, 1986. Reprint of the 1982 edition.
15. George Koletsos. Church-Rosser theorem for typed functionals. *Journal of Symbolic Logic*, 50:782–790, 1985.
16. Jean-Louis Krivine. *Lambda-calcul Types et modèles*. Masson, Paris, 1990.
17. Jean-Louis Krivine. *Lambda-calculus, types and models*. Ellis Horwood, New York, 1993. Translated from the 1990 French original by René Cori.
18. Daniel Leivant. Typing and computational properties of lambda expressions. *Theoret. Comput. Sci.*, 44(1):51–68, 1986.
19. Per Martin-Löf. Lecture notes on domain interpretation of type theory. Programming Methodology Group, Workshop on the Semantics of Programming Languages, Chalmers University of Technology, 1983.
20. John 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.
21. John C. Mitchell. *Foundation for Programmimg Languages*. MIT Press, 1996.
22. Gordon D. Plotkin. Set-theoretical and other elementary models of the λ-calculus. *Theoret. Comput. Sci.*, 121(1-2):351–409, 1993.
23. Garrel 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.
24. Dana 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.
25. Dana 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, Berlin, 1975.
26. Dana S. Scott. Domains for denotational semantics. In M.Nielsen and E.M.Schmidt, editors, *Automata, Languages and Programming*, pages 577–613. Springer, Berlin, 1982.
27. Richard Statman. Logical relations and the typed λ-calculus. *Information and Control*, 65:85–97, 1985.
28. William W. Tait. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic*, 32:198–212, 1967.
29. William 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. Springer, 1975.
30. Steffen van Bakel. Complete restrictions of the intersection type discipline. *Theoret. Comput. Sci.*, 102(1):135–163, 1992.