Available online at www.sciencedirect.com
SCIENCE DIRECT
Theoretical Computer Science 355 (2006) 108–126
Theoretical Computer Science
www.elsevier.com/locate/tcs

Intersection types and lambda models

Fabio Alessi[a], Franco Barbanera[b,1], Mariangiola Dezani-Ciancaglini[c,*,2]

[a] *Dip. di Matematica e Informatica, Università di Udine, Via delle Scienze 208, 33100 Udine, Italy*
[b] *Dip. di Matematica e Informatica, Università di Catania, Viale A.Doria 6, 95125 Catania, Italy*
[c] *Dip. di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy*

Abstract

Invariance of interpretation by β-conversion is one of the minimal requirements for any standard model for the λ-calculus. With the intersection-type systems being a general framework for the study of semantic domains for the λ-calculus, the present paper provides a (syntactic) characterisation of the above mentioned requirement in terms of characterisation results for intersection-type assignment systems.

Instead of considering conversion as a whole, reduction and expansion will be considered separately. Not only for usual computational rules like β, η, but also for a number of relevant restrictions of those. Characterisations will be also provided for (intersection) filter structures that are indeed λ-models.

Keywords: λ-calculus; Intersection types; λ-models; β- and η-reduction/expansion

1. Introduction

In the λ-calculus, the computational model at the basis of the functional programming paradigm, the basic step of computation is usually identified with the notion of β-reduction:

$$(\lambda x.M)N \to_\beta M[x := N].$$

Whereas, like any "computation rule", its role is (roughly) to make more *explicit* the "information" represented by a λ-term, such information, intuitively, must not be modified by the computational process embodied by the rule itself. That is why any classical notion of denotational interpretation for the λ-calculus has to respect minimal requisites w.r.t. the operational interpretation of the calculus. As a matter of fact, any classical denotational semantics for the λ-calculus must be required *sound*, that is it must interpret any two *convertible* terms with the very same information (denotational value).

* Corresponding author. Tel.: +39 011 670 6732; fax: +39 011 751603.
E-mail addresses: alessi@dimi.uniud.it (F. Alessi), barba@dmi.unict.it (F. Barbanera), dezani@di.unito.it (M. Dezani-Ciancaglini).

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

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.01.004

The soundness requirement for denotational models for the λ-calculus is, at large, the context of the present paper. In particular, we address the study of this requirement at a deeper level, that is, we "decompose" it into two separate requirements to be investigated individually: one concerning β-reduction alone and one concerning β-expansion.

(a) For any N s.t. $M \to_\beta N$, $[\![M]\!] = [\![N]\!]$,
(b) For any N s.t. $N \to_\beta M$, $[\![M]\!] = [\![N]\!]$,

where $[\![M]\!]$ represents the denotational interpretation of the term M in a given domain.

Due to the large variety of possible denotational models for the λ-calculus, such an investigation cannot be successfully undertaken unless we manage to identify a finitary and natural framework where most of the models proposed in the literature could be "embedded" and analysed.

Type assignment systems for the untyped λ-calculus with intersection types are definitely a framework with the qualities we are looking for: they form a class of type assignment systems which allow to express, in a natural and finitary way, many of the most important *denotational* properties of terms (as a matter of fact, also many relevant *operational* properties can be characterised by means of intersection types).

Indeed, intersection types are a powerful tool for both the analysis and synthesis of λ-models (see e.g. [9,11,18,24,23, 27,15,6] and the references there): on the one hand, intersection-type disciplines provide finitary inductive definitions of interpretation of λ-terms in models. On the other hand, they are suggestive for the shape the domain model has to have in order to exhibit certain properties (see e.g. [11,24,5,7,17,14]).

Intersection types can be also viewed as a restriction of the domain theory in logical form, see [1], to the special case of modelling pure λ-calculus by means of ω-algebraic complete lattices. Many properties of these models can be proved using this paradigm, which goes back to Stone duality.

Different finitary characterisations of models for the λ-calculus can be obtained by introducing specific constants, typing rules and type preorders in the basic intersection-type assignment system. An element of a particular domain, representing the denotational meaning of a term M, comes out to correspond to the set of types that can be inferred for M.

It is then clear that, in the framework of intersection-type systems, the study of the requirements (a) and (b) mentioned above can be fully formalised in terms of *typing invariance*, that is, in type theory terminology, by the so-called Subject Reduction and Expansion properties. Hence, particular (syntactic) characterisations of those domains where the requirement (a) (resp., (b)) is met, can be achieved by isolating necessary and sufficient conditions enabling a type system to enjoy the property of Subject Reduction (resp., Subject Expansion). One of the main results of the present paper consists in a number of such necessary and sufficient conditions for these properties.

It is worth noticing that many restrictions of the β-rule have been devised in the literature with the aim of formalising particular sorts of computations. Interesting examples of such restrictions are the rule of Plotkin's λ_v-calculus [26], the rule of $\lambda\mathbf{I}$-calculus [13] and the rule of $\lambda\mathbf{K}\mathsf{N}$-calculus [23]. In this paper, we shall prove our results also for the restricted notions of computations embodied by the above mentioned calculi.

β-conversion as a whole will be also taken into account in this paper, but from a rather broader perspective (forming the basis for further research): characterisation results will be in fact provided for those filter (intersection) structures that are also λ-models for the aforementioned calculi. Such results will profit from the characterisation results concerning Subject Reduction and Subject Expansion.

The extensionality property in the denotational semantics for the λ-calculus will be taken into account in terms of its syntactic formalisation: the η-rule. We shall show how to characterise the intersection-type systems enjoying Subject Reduction and Subject Expansion properties with respect to η-rule, as well as the filter structures that are extensional λ-models for the considered calculi.

This paper is structured as follows: in Section 2, we recall the definitions of intersection types and intersection-type preorders. We shall briefly recall the main systems proposed in the literature, in particular those related to the use of intersection types for denotational semantics. We shall also introduce conditions on type preorders to be used in our characterisation results. Section 3 discusses intersection-type assignment systems and their properties. Section 4 will contain our characterisation results concerning β- and η-reduction/expansion. Our characterisations of filter structure that are λ-models will be given in Section 5. Section 6 will provide a few remarks on possible further research on the arguments of the paper.

The present paper extends [3,4] providing all the omitted proofs in those preliminary versions.

2. Intersection-type languages and type preorders

In this section, we shall recall the main notions concerning intersection-type languages and type preorders.

Intersection types are syntactic objects built by closing a given set $\mathbb{C}$ of *type atoms* (constants) under the *function-type* constructor $\to$, and the *intersection-type* constructor $\cap$.

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

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

Notation: Upper case Roman letters i.e. $A, B, \ldots$, will denote arbitrary types. When writing intersection types we shall use the following convention: the constructor $\cap$ takes precedence over the constructor $\to$, and $\to$ associates to the right. For example $(A \to B \to C) \cap A \to B \to C \equiv ((A \to (B \to C)) \cap A) \to (B \to C)$.

In this paper, we shall be concerned with several different intersection-type languages arising from taking different sets of type atoms, depending on which typing invariance properties we want to capture. Typical choices for the set of type atoms are $\mathbb{C}_\infty$, a countable set of constants, or finite sets like $\{\Omega, \varphi, \omega\}$ or $\{\nu\}$.

Most of the expressive power of intersection-type languages comes from the fact that they are endowed with a *preorder relation*, $\leqslant$, which induces, on the set of types, the structure of a meet semi-lattice with respect to $\cap$. This appears natural when we think of types as sets of denotations, and interpret $\cap$ as set-theoretic intersections and $\leqslant$ as set inclusion.

Definition 2 (*Intersection-type preorder*). An *intersection-type preorder* $\Sigma = (\mathbb{C}^\Sigma, \leqslant_\Sigma)$ is a binary relation $\leqslant_\Sigma$ on $\mathbb{T}(\mathbb{C}^\Sigma)$ satisfying the set of axioms and rules of Fig. 1.

Notation: We shall write $A \sim_\Sigma B$ when both $A \leqslant_\Sigma B$ and $B \leqslant_\Sigma A$.

Axiom (Ω) states that the type preorders containing the constant Ω have Ω itself as top element. This is particularly meaningful when used in combination with the Ω-type assignment systems, which essentially treat Ω as the universal type of all λ-terms (see Definition 14).

Axiom (ν) states that ν is above any arrow type. This axiom agrees with the ν-type assignment systems, which treat ν as the universal type of all λ-abstractions (see Definition 16). Note that the role of ν may be played by the type $\Omega \to \Omega$, when Ω is in $\mathbb{C}$. For this reason, it is of no use to have at the same time ν and Ω. Hence, we impose, as a pragmatic rule, that these two constants do not occur together in any $\mathbb{C}$.

Note that associativity and commutativity of $\cap$ (as always modulo $\sim$) follow easily from the above axioms and rules. For instance, commutativity descends from (idem), (incl_L) and (incl_R) and (mon), as follows:

$$A \cap B \leqslant (A \cap B) \cap (A \cap B) \leqslant B \cap A.$$

Since $\cap$ is commutative and associative, we shall write $\bigcap_{i \leqslant n} A_i$ for $A_1 \cap \cdots \cap A_n$. Similarly, we shall write $\bigcap_{i \in I} A_i$, where I denotes always a finite set. Moreover, we convene that $\bigcap_{i \in \emptyset} A_i$ is Ω when $\Omega \in \mathbb{C}$ and we forbid intersections on the empty set when $\Omega \notin \mathbb{C}$.

Remark 3. It is not required that $\sim$ be congruent with the constructor $\to$. For many type preorders this will be implied by the extra axiom (η) or ($\eta^\sim$) (see Fig. 2).

$$\text{(refl)}\ A \leq A \qquad \text{(idem)}\ A \leq A \cap A$$

$$\text{(incl}_L\text{)}\ A \cap B \leq A \qquad \text{(incl}_R\text{)}\ A \cap B \leq B$$

$$\text{(mon)}\ \frac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'} \qquad \text{(trans)}\ \frac{A \leq B \quad B \leq C}{A \leq C}$$

$$(\Omega)\ \text{if } \Omega \in \mathbb{C} \quad A \leq \Omega \qquad (\nu)\ \text{if } \nu \in \mathbb{C} \quad A \to B \leq \nu$$

Fig. 1. Basic axioms and rules of type preorders.

(Ω-η)	$\Omega \leq \Omega \to \Omega$	($\to$-$\cap$)	$(A \to B) \cap (A \to C) \leq A \to B \cap C$
(Ω-*lazy*)	$A \to B \leq \Omega \to \Omega$	($\to$-$\cap^{\sim}$)	$(A \to B) \cap (A \to C) \sim A \to B \cap C$
(η)	$\dfrac{A' \leq A \quad B \leq B'}{A \to B \leq A' \to B'}$	($\eta^{\sim}$)	$\dfrac{A' \sim A \quad B \sim B'}{A \to B \sim A' \to B'}$
(ω-*Scott*)	$\Omega \to \omega \sim \omega$	(ω-*Park*)	$\omega \to \omega \sim \omega$
($\omega\varphi$)	$\omega \leq \varphi$	($\varphi \to \omega$)	$\varphi \to \omega \sim \omega$
($\omega \to \varphi$)	$\omega \to \varphi \sim \varphi$	(I)	$(\varphi \to \varphi) \cap (\omega \to \omega) \sim \omega$

Fig. 2. Possible axioms and rules concerning $\leqslant$.

All the type preorders considered so far in the literature are defined for languages over finite or countable sets of atoms and are "generated" by recursive sets $\bigtriangledown$ of axioms and rules of the shape $A \leqslant B$ (where $\bigtriangledown$ it is said to generate $\leqslant$ when $A \leqslant B$ holds if and only if it can be derived from the axioms and rules of $\bigtriangledown$ together with those in Definition 2). Such generated preorders have been referred to as *type theories*. We shall denote them by $\Sigma^{\bigtriangledown} = (\mathbb{C}^{\bigtriangledown}, \leqslant_{\bigtriangledown})$.

Note that there are only countably many possible $\bigtriangledown$; hence, there are uncountably many preorders which cannot be represented this way. Note also that the correspondence $\bigtriangledown \mapsto \leqslant_{\bigtriangledown}$ is not injective.

In this paper, we try to be as general as possible, sticking to our notion of type preorder which indeed extends the notion of type preorders usually considered in the literature, where rules (Ω) and (ν) are not taken into account and are instead postulated inside the recursive sets generating the type preorder.

Fig. 2 shows a list of special purpose axioms and rules which have been considered in the literature, and which we shall briefly discuss in the following.

The meaning of axioms and rules of Fig. 2 can be grasped if we take types to denote subsets of a domain of discourse and we look at $\to$ as the function space constructor in the light of Curry–Scott semantics, see [29]. Thus, the type $A \to B$ denotes the set of *total* functions which map each element of A into an element of B.

Since Ω represents the maximal element, i.e. the whole universe, $\Omega \to \Omega$ is the set of functions which applied to an arbitrary element return again an arbitrary element. Thus, axiom (Ω-η) expresses the fact that all the objects in our domain of discourse are total functions, i.e. that Ω is equal to $\Omega \to \Omega$ [9]. If now we want to capture only those terms which truly represent functions, as we do for example in the lazy λ-calculus, we cannot assume axiom (Ω-η). One still may postulate the weaker property (Ω-*lazy*) to make all functions total [2]. It simply says that an element which is a function, because it maps A into B, maps also the whole universe into itself.

The intended interpretation of arrow types motivates axiom ($\to$-$\cap$), which implies that if a function maps A into B, and the same function maps also A into C, then, actually, it maps the whole A into the intersection of B and C (i.e. into $B \cap C$), see [9].

Rule (η) is also very natural in view of the set-theoretic interpretation. It implies that the arrow constructor is contravariant in the first argument and covariant in the second one. It is clear that if a function maps A into B, and we take a subset A' of A and a superset B' of B, then this function will map also A' into B', see [9].

The rules ($\to$-$\cap^{\sim}$) and ($\eta^{\sim}$) are similar to the rules ($\to$-$\cap$) and (η). They capture properties of the graph models for the untyped λ-calculus, see [27,19].

The remaining axioms express peculiar properties of D_{∞}-like inverse limit models, see [9,12,11,24,22,14].

We can introduce now a list of significant intersection type preorders which have been extensively considered in the literature. All these preorders have been introduced mainly to obtain corresponding filter models of (restricted) λ-calculi, as we shall discuss in Section 5. The order is logical, rather than historical, and some references define the models, others deal with the corresponding filter models: [30,9–11,17–19,23–25,2,28,27].

These preorders are of the form $\Sigma^{\bigtriangledown} = (\mathbb{C}^{\bigtriangledown}, \leqslant_{\bigtriangledown})$, with various different names $\bigtriangledown$, picked for mnemonic reasons. In Fig. 3, for each preorder $\Sigma^{\bigtriangledown}$ we list its set $\mathbb{C}^{\bigtriangledown}$ of constants and its set $\bigtriangledown$ of extra axioms and rules taken from Fig. 2. Here, $\mathbb{C}_{\infty}$ is an infinite set of fresh atoms (i.e. different from Ω, ν, φ, ω).

We define two conditions on type preorders to be used in our characterisation results for rule β.

Definition 4 (*Beta and ν-sound preorders*).

(1) A type preorder Σ is *beta* iff for all sets of indexes I, and all types A_i, B_i, C, D in $\mathbb{T}(\mathbb{C}^{\Sigma})$:

$$\bigcap_{i \in I} (A_i \to B_i) \leqslant_{\Sigma} C \to D \quad \Leftrightarrow \quad \bigcap_{i \in J} B_i \leqslant_{\Sigma} D \quad \text{where } J = \{i \in I \mid C \leqslant A_i\}.$$

$$
\begin{array}{llll}
\mathbb{C}^{\mathcal{B}a} & = \mathbb{C}_\infty & \mathcal{B}a & = \emptyset \\
\mathbb{C}^{\mathcal{CDV}} & = \mathbb{C}_\infty & \mathcal{CDV} & = \{(\to\text{-}\cap), (\eta)\} \\
\mathbb{C}^{\mathcal{HL}} & = \{\varphi, \omega\} & \mathcal{HL} & = \mathcal{CDV} \cup \{(\omega\varphi), (\varphi \to \omega), (\omega \to \varphi)\} \\
\mathbb{C}^{\mathcal{HR}} & = \{\varphi, \omega\} & \mathcal{HR} & = \mathcal{CDV} \cup \{(\omega\varphi), (\omega \to \varphi), (I)\} \\
\mathbb{C}^{\mathcal{EHR}} & = \{\nu\} & \mathcal{EHR} & = \mathcal{CDV} \\
\mathbb{C}^{\mathcal{AO}} & = \{\Omega\} & \mathcal{AO} & = \mathcal{CDV} \cup \{(\Omega\text{-}lazy)\} \\
\mathbb{C}^{\mathcal{BCD}} & = \{\Omega\} \cup \mathbb{C}_\infty & \mathcal{BCD} & = \mathcal{CDV} \cup \{(\Omega\text{-}\eta)\} \\
\mathbb{C}^{\mathcal{Sc}} & = \{\Omega, \omega\} & \mathcal{Sc} & = \mathcal{BCD} \cup \{(\omega\text{-}Scott)\} \\
\mathbb{C}^{\mathcal{P}a} & = \{\Omega, \omega\} & \mathcal{P}a & = \mathcal{BCD} \cup \{(\omega\text{-}Park)\} \\
\mathbb{C}^{\mathcal{CDZ}} & = \{\Omega, \varphi, \omega\} & \mathcal{CDZ} & = \mathcal{HL} \cup \mathcal{BCD} \\
\mathbb{C}^{\mathcal{P}l} & = \{\Omega, \varphi\} & \mathcal{P}l & = \{(\eta^\sim)\} \\
\mathbb{C}^{\mathcal{E}n} & = \{\Omega\} \cup \mathbb{C}_\infty & \mathcal{E}n & = \mathcal{P}l \cup \{(\to\text{-}\cap^\sim), (\Omega\text{-}\eta)\} \\
\mathbb{C}^{\mathcal{DHM}} & = \{\Omega, \varphi, \omega\} & \mathcal{DHM} & = \mathcal{BCD} \cup \{(\omega\varphi), (\omega\text{-}Scott), (\omega \to \varphi)\}
\end{array}
$$

Fig. 3. Particular atoms, axioms and rules.

(2) A type preorder Σ is *v-sound* iff for all $A, B \in \mathbb{T}(\mathbb{C}^\Sigma)$:

$$v \not\sim_\Sigma A \to B.$$

A few comments on the previous definition. In the definition of beta preorders, if J is empty and $\Omega \in \mathbb{C}^\Sigma$ we get $\Omega \sim_\Sigma D$. Instead, by assumption, J can never be empty when $\Omega \notin \mathbb{C}^\Sigma$. If we look at $\cap$ as representing join, and arrow types as representing step functions, then the condition for a type preorder of being beta is exactly the relation which holds between sups of step functions [20].

The v-sound condition is used both to prevent v from being a redundant type and to avoid assigning too many types to a λ-abstraction (assigning v amounts exactly to discriminating an abstraction and nothing more). Note that Σ is trivially v-sound when $v \notin \mathbb{C}^\Sigma$.

When $\Sigma = \Sigma^\bigtriangledown$, for some $\bigtriangledown$, it is usually possible to prove the above defined conditions by induction on the derivation that shows that two given types are in the preorder relation. The following notion of *strong beta* is handy when proving that some of the type preorders of Fig. 3 are beta. A type preorder $\Sigma^\bigtriangledown$ is strong beta when its set $\bigtriangledown$ of axioms and rules contains $\mathcal{BCD}$ and a set $\bigtriangledown^-$ of axioms with suitable properties.

Definition 5 (*Strong beta preorders*). A type preorder $\Sigma^\bigtriangledown$ is *strong beta* if $\bigtriangledown = \mathcal{BCD} \cup \bigtriangledown^-$ and:

(1) $\bigtriangledown^-$ contains no rule and only axioms of one of the following two shapes:
- $\psi \leqslant \psi'$,
- $\psi \sim \bigcap_{i \in I} (\psi_i^{(1)} \to \psi_i^{(2)})$,

where $\psi, \psi', \psi_i^{(1)}, \psi_i^{(2)} \in \mathbb{C}^\bigtriangledown$, and $\psi, \psi', \psi_i^{(2)} \not\equiv \Omega$ for all $i \in I$;

(2) for each $\psi \in \mathbb{C}^\bigtriangledown$ such that $\psi \not\equiv \Omega$ there is exactly one axiom in $\bigtriangledown^-$ of the shape $\psi \sim \bigcap_{i \in I} (\psi_i^{(1)} \to \psi_i^{(2)})$;

(3) let $\bigtriangledown^-$ contain $\psi \sim \bigcap_{i \in I} (\psi_i^{(1)} \to \psi_i^{(2)})$ and $\psi' \sim \bigcap_{j \in J} (\psi_j'^{(1)} \to \psi_j'^{(2)})$. Then $\bigtriangledown^-$ contains also $\psi \leqslant \psi'$ iff for each $j \in J$ there exists $i \in I$ such that $\psi_j'^{(1)} \leqslant \psi_i^{(1)}$ and $\psi_i^{(2)} \leqslant \psi_j'^{(2)}$ are both in $\bigtriangledown^-$.

For example, the preorders $\Sigma^{\mathcal{HL}}$, $\Sigma^{\mathcal{Sc}}$, $\Sigma^{\mathcal{P}a}$, $\Sigma^{\mathcal{CDZ}}$, $\Sigma^{\mathcal{HR}}$ and $\Sigma^{\mathcal{DHM}}$ are strong beta.

Lemma 6. *Each strong beta-type preorder is beta.*

Proof. We shall denote elements of $\mathbb{C}^\bigtriangledown$ by $\psi, \xi, \varphi, \zeta$ (possibly with indexes). By assumption, for each constant $\psi \in \mathbb{C}^\bigtriangledown$ there is exactly *one* axiom stating that ψ is equivalent to an intersection of arrow types. We denote such an intersection

by $\bigcap_{l\in L^{(\psi)}}(\xi_l^{(\psi)}\to\zeta_l^{(\psi)})$. Moreover, note that the most general form of an intersection type is a finite intersection of arrow types and type constants. We can prove two statements by simultaneous induction on the definition of $\leqslant$, the first of which implies the beta condition:

- if $(\bigcap_{i\in I}(A_i\to B_i))\cap(\bigcap_{h\in H}\psi_h)\leqslant_\triangledown(\bigcap_{j\in J}(C_j\to D_j))\cap(\bigcap_{k\in K}\varphi_k)$, then $\forall j\in J.\,(\bigcap_{i\in I'}B_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(\psi_h)'}}\zeta_l^{(\psi_h)}))\leqslant_\triangledown D_j$ where $I'=\{i\in I\mid C_j\leqslant_\triangledown A_i\}$, $L^{(\psi_h)'}=\{l\in L^{(\psi_h)}\mid C_j\leqslant_\triangledown\xi_l^{(\psi_h)}\}$, $H'=\{h\in H\mid L^{(\psi_h)'}\neq\emptyset\}$;
- if $(\bigcap_{i\in I}(A_i\to B_i))\cap(\bigcap_{h\in H}\psi_h)\leqslant_\triangledown(\bigcap_{j\in J}(C_j\to D_j))\cap(\bigcap_{k\in K}\varphi_k)$, then $\forall k\in K,\,m\in L^{(\varphi_k)}.\,(\bigcap_{i\in I'}B_i)\cap(\bigcap_{h\in H'}(\bigcap_{l\in L^{(\psi_h)'}}\zeta_l^{(\psi_h)}))\leqslant_\triangledown\zeta_m^{(\varphi_k)}$ where $I'=\{i\in I\mid\xi_m^{(\varphi_k)}\leqslant_\triangledown A_i\}$, $L^{(\psi_h)'}=\{l\in L^{(\psi_h)}\mid\xi_m^{(\varphi_k)}\leqslant_\triangledown\xi_l^{(\psi_h)}\}$, $H'=\{h\in H\mid L^{(\psi_h)'}\neq\emptyset\}$. □

Proposition 7.
(1) *Type preorders of Fig.* 3 *are beta.*
(2) *Type preorders of Fig.* 3 *are v-sound.*

Proof. (1) For $\triangledown\in\{\mathcal{B}a,\mathcal{CDV},\mathcal{EHR},\mathcal{AO},\mathcal{BCD},\mathcal{P}l,\mathcal{E}n\}$ we can prove, by induction on the definition of $\leqslant_\triangledown$, that

$$\left(\bigcap_{i\in I}(A_i\to B_i)\right)\cap\left(\bigcap_{h\in H}\psi_h\right)\leqslant_\triangledown\left(\bigcap_{j\in J}(C_j\to D_j)\right)\cap\left(\bigcap_{k\in K}\xi_k\right)\ \Rightarrow\ \bigcap_{i\in I'}B_i\leqslant_\triangledown D_j,$$

where $I'=\{i\in I\mid C_j\leqslant_\triangledown A_i\}$.

The preorders $\Sigma^\triangledown$ for $\triangledown\in\{\mathcal{HL},\mathcal{S}c,\mathcal{P}a,\mathcal{CDZ},\mathcal{HR},\mathcal{DHM}\}$ are beta by Lemma 6.

(2) For $\Sigma^{\mathcal{EHR}}$ one can easily show, by induction on $\leqslant_{\mathcal{EHR}}$, that $v\leqslant_{\mathcal{EHR}}A$ implies that A is an intersection of v. □

Example 8. An example of a non-beta preorder is $\Sigma^\diamondsuit$, defined by $\mathbb{C}^\diamondsuit=\{\Omega,\diamondsuit,\heartsuit\}$ and $\diamondsuit=\mathcal{BCD}\cup\{(\diamondsuit)\}$, where

$$(\diamondsuit)\quad A\leqslant A[\diamondsuit:=\heartsuit].$$

$\Sigma^\diamondsuit$ is not beta, since $\diamondsuit\to\diamondsuit\leqslant_\diamondsuit\heartsuit\to\heartsuit$, but $\heartsuit\not\leqslant_\diamondsuit\diamondsuit$.

Notation: We write "the type preorder Σ validates $\triangledown$" to mean that all axioms and rules of $\triangledown$ are admissible in Σ.

In order to characterise the invariance of typing under η-expansion, we need to introduce a further condition on type preorders, which essentially says that each atomic type either is greater than or equal to a type which can be deduced for all terms which are abstractions (see Definition 16), or it is between two intersections of "strictly related" arrow types, as specified in the following definition.

Definition 9 (*Eta preorders*). A type preorder Σ is *eta* iff for all $\psi\in\mathbb{C}^\Sigma$ one of these two conditions hold:

- $v\leqslant_\Sigma\psi$;
- there exist non-empty families of types $\{A_i,B_i\}_{i\in I}$, $\{D_{i,j},E_{i,j}\}_{j\in J_i}$ in $\mathbb{T}(\mathbb{C}^\Sigma)$ such that

$$\bigcap_{i\in I}(A_i\to B_i)\leqslant_\Sigma\psi\leqslant_\Sigma\bigcap_{i\in I'}\left(\bigcap_{j\in J_i}(D_{i,j}\to E_{i,j})\right)\quad\&\quad\forall i\in I'.\,A_i\leqslant_\Sigma\bigcap_{j\in J_i}D_{i,j}\quad\&\quad\bigcap_{j\in J_i}E_{i,j}\leqslant_\Sigma B_i,$$

where $I'=\{i\in I\mid B_i\not\sim_\Sigma\Omega\}$.

It is easy to verify that if either $\Omega\notin\mathbb{C}^\Sigma$ and Σ validates $\mathcal{CDV}$ or $\Omega\in\mathbb{C}^\Sigma$ and Σ validates $\mathcal{AO}$, then the condition of the above definition simplifies to the requirement that all atomic types are either greater than v or greater than $\Omega\to\Omega$, or they are equivalent to a suitable intersection of arrow types, namely

$$\forall\psi\in\mathbb{C}^\Sigma.\ v\leqslant_\Sigma\psi\ \text{ or }\ \Omega\to\Omega\leqslant_\Sigma\psi\ \text{ or }\ \exists I,\quad\{A_i,B_i\}_{i\in I}.\ \bigcap_{i\in I}(A_i\to B_i)\sim_\Sigma\psi.$$

The following proposition singles out all type preorders of Fig. 3 which are eta: the proof is trivial.

Proposition 10. *If* $\bigtriangledown \in \{\mathcal{HL}, \mathcal{HR}, \mathcal{EHR}, \mathcal{AO}, \mathcal{Sc}, \mathcal{Pa}, \mathcal{CDZ}, \mathcal{DHM}\}$, *then* $\Sigma^{\bigtriangledown}$ *is a eta preorder.*

3. Intersection-type assignments

We are now ready to introduce the crucial notion of *intersection-type assignment system*. First, we need a few preliminary definitions.

Definition 11.
(1) A Σ-*basis* is a set of statements of the shape $x : B$, where $B \in \mathbb{T}(\mathbb{C}^{\Sigma})$. All term variables occurring in a Σ-basis are distinct.
(2) An *intersection-type assignment system* relative to Σ, denoted by $\lambda\cap^{\Sigma}$, is a formal system for deriving judgements of the form $\Gamma \vdash^{\Sigma} M : A$, where the *subject* M is a λ-term, the *predicate* A is in $\mathbb{T}(\mathbb{C}^{\Sigma})$, and Γ is a Σ-basis.

We shall consider λ-terms up to α-conversion and we shall assume the Barendregt convention on variables [8] to be fulfilled. The Barendregt convention for judgments $\Gamma \vdash^{\Sigma} M : A$ implies that variables occurring in the Σ-basis Γ cannot occur bound in the term M.

Notation: If Γ is a Σ-basis then $x \in \Gamma$ is short for $(x : A) \in \Gamma$ for some A.
If Γ is a Σ-basis and $A \in \mathbb{T}(\mathbb{C}^{\Sigma})$ then $\Gamma, x : A$ is short for $\Gamma \cup \{x : A\}$ when $x \notin \Gamma$.
When $\Sigma = \Sigma^{\bigtriangledown}$ we shall denote $\lambda\cap^{\Sigma}$ and $\vdash^{\Sigma}$ by $\lambda\cap^{\bigtriangledown}$ and $\vdash^{\bigtriangledown}$, respectively.
Various type assignment systems can be defined, each of them parametrized with a particular type preorder Σ. The simplest system is given in the following definition.

Definition 12 (*Basic-type assignment system*). Given a type preorder Σ, the axioms and rules of the *basic-type assignment system*, denoted by $\lambda\cap^{\Sigma}_{\mathcal{B}}$, for deriving judgements $\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A$, are shown in Fig. 4.

Example 13. Self-application can be easily typed in $\lambda\cap^{\Sigma}_{\mathcal{B}}$, as follows:

$$\dfrac{\dfrac{\dfrac{x{:}(A \to B) \cap A \vdash^{\Sigma}_{\mathcal{B}} x{:}(A \to B) \cap A}{x{:}(A \to B) \cap A \vdash^{\Sigma}_{\mathcal{B}} x{:}A \to B}\,(\leqslant) \quad \dfrac{x{:}(A \to B) \cap A \vdash^{\Sigma}_{\mathcal{B}} x{:}(A \to B) \cap A}{x{:}(A \to B) \cap A \vdash^{\Sigma}_{\mathcal{B}} x{:}A}\,(\leqslant)}{x{:}(A \to B) \cap A \vdash^{\Sigma}_{\mathcal{B}} xx{:}B}\,(\to \mathrm{E})}{\vdash^{\Sigma}_{\mathcal{B}} \lambda x.xx : (A \to B) \cap A \to B}\,(\to \mathrm{I})$$

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

Definition 14 (*Ω-type assignment system*). Given a type preorder Σ with $\Omega \in \mathbb{C}^{\Sigma}$, the axioms and rules of the *Ω-type assignment system* (denoted $\lambda\cap^{\Sigma}_{\Omega}$), for deriving judgements of the form $\Gamma \vdash^{\Sigma}_{\Omega} M : A$, are those of the basic one, plus the axiom

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

$$(\mathbf{Ax})\quad \Gamma, x{:}A \vdash^{\Sigma}_{\mathcal{B}} x{:}A$$

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

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

Fig. 4. The axioms and rules of the basic-type assignment system.

Example 15. Also non-strongly normalising terms can be typed in $\lambda\cap^{\Sigma}_{\Omega}$ even with a type different from Ω. Note the usage of the axiom $(Ax\text{-}\Omega)$. Let $\Delta \equiv \lambda x.xx$.

$$\dfrac{\dfrac{\dfrac{x{:}A,\, y{:}\Omega \vdash^{\Sigma}_{\Omega} x : A}{y{:}\Omega \vdash^{\Sigma}_{\Omega} \lambda x.x : A \to A}(\to \mathrm{I})}{\vdash^{\Sigma}_{\Omega} \lambda yx.x : \Omega \to A \to A}(\to \mathrm{I}) \qquad \vdash^{\Sigma}_{\Omega} \Delta\Delta : \Omega}{\vdash^{\Sigma}_{\Omega} (\lambda yx.x)(\Delta\Delta) : A \to A}(\to \mathrm{E}).$$

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

Definition 16 (*ν-type assignment system*). Given a type preorder Σ with $\nu{\in}\mathbb{C}^{\Sigma}$, the axioms and rules of the *ν-type assignment system* (denoted $\lambda\cap^{\Sigma}_{\nu}$), for deriving judgements of the form $\Gamma \vdash^{\Sigma}_{\nu} M : A$, are those of the basic one, plus the axiom

$$(Ax\text{-}\nu)\quad \Gamma \vdash^{\Sigma}_{\nu} \lambda x.M : \nu.$$

Example 17. Axiom $(Ax\text{-}\nu)$ allows again to type non-strongly normalising terms. Note that the term of Example 15 is not typable in $\lambda\cap^{\mathcal{EHR}}_{\nu}$, as proved in [18].

$$\dfrac{\dfrac{\dfrac{x{:}A,\, y{:}\nu \vdash^{\Sigma}_{\nu} x : A}{y{:}\nu \vdash^{\Sigma}_{\nu} \lambda x.x : A \to A}(\to \mathrm{I})}{\vdash^{\Sigma}_{\nu} \lambda yx.x : \nu \to A \to A}(\to \mathrm{I}) \qquad \vdash^{\Sigma}_{\nu} \lambda z.\Delta\Delta : \nu}{\vdash^{\Sigma}_{\nu} (\lambda yx.x)(\lambda z.\Delta\Delta) : A \to A}(\to \mathrm{E}).$$

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

Notation: In the following, $\lambda\cap^{\Sigma}$ will range over $\lambda\cap^{\Sigma}_{\mathcal{B}}$, $\lambda\cap^{\Sigma}_{\Omega}$ and $\lambda\cap^{\Sigma}_{\nu}$. More precisely, we shall assume that $\lambda\cap^{\Sigma}$ stands for $\lambda\cap^{\Sigma}_{\Omega}$ whenever $\Omega{\in}\mathbb{C}^{\Sigma}$, for $\lambda\cap^{\Sigma}_{\nu}$ whenever $\nu{\in}\mathbb{C}^{\Sigma}$, and for $\lambda\cap^{\Sigma}_{\mathcal{B}}$ otherwise. Similarly, for $\vdash^{\Sigma}$. If there is no danger of confusion, we write simply $\vdash$ for $\vdash^{\Sigma}$.

The subterm property does not hold in general for $\lambda\cap^{\Sigma}_{\nu}$. In fact, $\lambda x.M$ is typable also when M is not typable. Moreover, in $\lambda\cap^{\Sigma}_{\Omega}$ and $\lambda\cap^{\Sigma}_{\nu}$, a judgement $\Gamma \vdash^{\Sigma} M : A$ does not imply $FV(M) \subseteq \Gamma$.

One of the most interesting features of intersection-type systems is that of enabling precise characterisation results of many important sets of λ-terms, among which the one of strongly normalising terms. Such a result is stated in the following theorem and it will be used in the next section (for a proof see [17]). [3]

Theorem 18 (*Characterisation of strongly normalising terms*). *A λ-term M is strongly normalising iff for all type preorders Σ, there exist $A{\in}\mathbb{T}(\mathbb{C}^{\Sigma})$ and a Σ-basis Γ such that $\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A$.*

We end this subsection by defining the union between Σ-basis which requires some care in the presence of the intersection type constructor.

Definition 19.

$$\begin{aligned}\Gamma_1 \uplus \Gamma_2 = {} & \{(x{:}A) \mid (x{:}A){\in}\Gamma_1 \text{ and } x \notin \Gamma_2\} \\ & \cup \{(x{:}A) \mid (x{:}A){\in}\Gamma_2 \text{ and } x \notin \Gamma_1\} \\ & \cup \{(x{:}A_1{\cap}A_2) \mid (x{:}A_1){\in}\Gamma_1 \text{ and } (x{:}A_2){\in}\Gamma_2\}.\end{aligned}$$

[3] The type systems considered in [17] are induced by type theories instead of type preorders, but the arguments given there to show the characterisation of strongly normalising terms extend without changes to type preorders.

In the rest of this section, we shall introduce a few relevant properties of intersection types, needed for our characterisation results in the following section.

3.1. Admissible rules

Many interesting type assignment rules can be proved to be admissible.

Proposition 20 (*Admissible rules*). *For any type preorder Σ, the following rules are admissible in the intersection-type assignment system $\lambda\cap^{\Sigma}$.*

$$(\cap\mathrm{E}_l)\quad \frac{\Gamma\vdash^{\Sigma} M{:}A\cap B}{\Gamma\vdash^{\Sigma} M{:}A} \qquad (\cap\mathrm{E}_r)\quad \frac{\Gamma\vdash^{\Sigma} M{:}A\cap B}{\Gamma\vdash^{\Sigma} M{:}B}$$

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

$$(\mathrm{C})\quad \frac{\Gamma,x{:}B\vdash^{\Sigma} M{:}A \quad \Gamma\vdash^{\Sigma} N{:}B}{\Gamma\vdash^{\Sigma} M[x:=N]{:}A} \qquad (\leqslant\mathrm{L})\quad \frac{\Gamma,x{:}B\vdash^{\Sigma} M{:}A \quad C\leqslant_{\Sigma} B}{\Gamma,x{:}C\vdash^{\Sigma} M{:}A}$$

In the following, we shall freely use the rules of Proposition 20.

3.2. Generation lemmata

We introduce now a few properties enabling to "reverse" some of the rules of the type assignment systems $\lambda\cap^{\Sigma}$, so as to achieve some form of generation (or inversion) lemmata (see Theorems 21 and 22).

Such properties are not trivial. For instance, for the arrow elimination rule, in general, we can only say that when $\Gamma\vdash^{\Sigma} MN : A$, then there are a non-empty, finite set I and types B_i, C_i, such that for each $i\in I$, $\Gamma\vdash^{\Sigma} M : B_i\to C_i$, $\Gamma\vdash^{\Sigma} N : B_i$, and moreover $\bigcap_{i\in I} C_i\leqslant_{\Sigma} A$. Reasoning similarly on the rule $(\to\mathrm{I})$, one can conclude again that it cannot be reversed. More formally, we get the following theorem.

Notation: When we write "...assume $A\not\sim_{\Sigma}\Omega$..." we mean that this condition is always true when we deal with $\vdash^{\Sigma}_{\mathcal{B}}$ and $\vdash^{\Sigma}_{\nu}$, while it must be checked for $\vdash^{\Sigma}_{\Omega}$. Similarly, the condition $\nu\not\leqslant_{\Sigma} A$ must be checked just for $\vdash^{\Sigma}_{\nu}$.

Theorem 21 (*Generation Lemma I*). *Let Σ be a type preorder.*

(1) *Assume $A\not\sim_{\Sigma}\Omega$. Then $\Gamma\vdash^{\Sigma} MN : A$ iff $\Gamma\vdash^{\Sigma} M : B_i\to C_i$, $\Gamma\vdash^{\Sigma} N : B_i$ and $\bigcap_{i\in I} C_i\leqslant_{\Sigma} A$ for some non-empty set I and types $B_i, C_i\in\mathbb{T}(\mathbb{C}^{\Sigma})$.*

(2) *Assume $\nu\not\leqslant_{\Sigma} A$. Then $\Gamma\vdash^{\Sigma}\lambda x.M : A$ iff $\Gamma, x : B_i\vdash^{\Sigma} M : C_i$ and $\bigcap_{i\in I}(B_i\to C_i)\leqslant_{\Sigma} A$ for some non-empty set I and types $B_i, C_i\in\mathbb{T}(\mathbb{C}^{\Sigma})$.*

Proof. The proof of each $(\Leftarrow)$ is easy. So we only treat $(\Rightarrow)$.

(1) By induction on derivations. The only interesting case is when $A\equiv A_1\cap A_2$ and the last applied rule is $(\cap\mathrm{I})$:

$$(\cap\mathrm{I})\quad \frac{\Gamma\vdash^{\Sigma} MN : A_1 \quad \Gamma\vdash^{\Sigma} MN : A_2}{\Gamma\vdash^{\Sigma} MN : A_1\cap A_2}.$$

The condition $A\not\leqslant_{\Sigma}\Omega$ implies that we cannot have $A_1\sim_{\Sigma} A_2\sim_{\Sigma}\Omega$. We do the proof for $A_1\not\leqslant_{\Sigma}\Omega$ and $A_2\not\leqslant_{\Sigma}\Omega$, the other cases can be treated similarly. By induction, there are I, B_i, C_i, J, D_j, E_j such that

$$\begin{array}{lccc} \forall\, i\in I. & \Gamma\vdash^{\Sigma} M : B_i\to C_i, & \Gamma\vdash^{\Sigma} N : B_i, \\ \forall\, j\in J. & \Gamma\vdash^{\Sigma} M : D_j\to E_j, & \Gamma\vdash^{\Sigma} N : D_j, \\ \bigcap\limits_{i\in I} C_i\leqslant_{\Sigma} A_1 & \& & \bigcap\limits_{j\in J} E_j\leqslant_{\Sigma} A_2. \end{array}$$

So we are done, since $(\bigcap_{i\in I} C_i)\cap(\bigcap_{j\in J} E_j)\leqslant_{\Sigma} A$.

(2) The proof is very similar to the proof of (1). It is again by induction on derivations and again the only interesting case is when the last applied rule is (∩I):

$$(\cap\mathrm{I}) \quad \frac{\Gamma \vdash^{\Sigma} \lambda x.M : A_1 \quad \Gamma \vdash^{\Sigma} \lambda x.M : A_2}{\Gamma \vdash^{\Sigma} \lambda x.M : A_1 \cap A_2}.$$

The condition $v \not\leqslant_{\Sigma} A$ implies that we cannot have $v \leqslant_{\Sigma} A_1$ and $v \leqslant_{\Sigma} A_2$. We do the proof for $v \not\leqslant_{\Sigma} A_1$ and $v \not\leqslant_{\Sigma} A_2$. By induction, there are I, B_i, C_i, J, D_j, E_j such that

$$\forall i \in I.\ \Gamma, x{:}B_i \vdash^{\Sigma} M : C_i, \quad \forall j \in J.\ \Gamma, x{:}D_j \vdash^{\Sigma} M : E_j,$$
$$\bigcap_{i\in I}(B_i \to C_i) \leqslant_{\Sigma} A_1 \quad \& \quad \bigcap_{j\in J}(D_j \to E_j) \leqslant_{\Sigma} A_2.$$

So we are done, since $(\bigcap_{i\in I}(B_i \to C_i)) \cap (\bigcap_{j\in J}(D_j \to E_j)) \leqslant_{\Sigma} A$. The other two cases are easier. For instance, if $v \not\leqslant_{\Sigma} A_1$ and $v \sim_{\Sigma} A_2$, it is sufficient to take $\bigcap_{i\in I}(B_i \to C_i)$ above to conclude. □

Using the properties introduced in Definition 4, we can give now a rather powerful generation lemma for $\lambda\cap^{\Sigma}$, which is one of the essential ingredients for the proofs of our results. We use the notion of "validation" introduced at p. 9.

Special cases of this theorem have been previously proved in [9,12,11,24,18].

Theorem 22 (*Generation Lemma II*). *Let Σ be a type preorder.*
(1) *Assume $A \not\sim_{\Sigma} \Omega$. Then $\Gamma \vdash^{\Sigma} x : A$ iff $(x{:}B) \in \Gamma$ and $B \leqslant_{\Sigma} A$ for some $B \in \mathbb{T}(\mathbb{C}^{\Sigma})$.*
(2) *Assume $A \not\sim_{\Sigma} \Omega$ and let Σ validate $\mathcal{CDV}$. Then $\Gamma \vdash^{\Sigma} MN : A$ iff $\Gamma \vdash^{\Sigma} M : B \to A$ and $\Gamma \vdash^{\Sigma} N : B$ for some $B \in \mathbb{T}(\mathbb{C}^{\Sigma})$.*
(3) *Let Σ be v-sound and beta. Then $\Gamma \vdash^{\Sigma} \lambda x.M : B \to C$ iff $\Gamma, x{:}B \vdash^{\Sigma} M : C$.*

Proof. The proof of each (⇐) is easy. So we only treat (⇒).

(1) Easy by induction on derivations, since only the axioms (Ax), $(Ax\text{-}\Omega)$, and the rules (∩I), (⩽) can be applied. Note that the condition $A \not\leqslant_{\Sigma} \Omega$ implies that $\Gamma \vdash^{\Sigma} x : A$ cannot be obtained just using axioms $(Ax\text{-}\Omega)$.

(2) Let I, B_i, C_i be as in Theorem 21(1). Applying rule (∩I) to $\Gamma \vdash^{\Sigma} M : B_i \to C_i$ we can derive $\Gamma \vdash^{\Sigma} M : \bigcap_{i\in I}(B_i \to C_i)$, so by rule (⩽) we have $\Gamma \vdash^{\Sigma} M : \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$. In fact, by rule (η) and axiom $(\to\text{-}\cap)$ we get $\bigcap_{i\in I}(B_i \to C_i) \leqslant_{\Sigma} \bigcap_{i\in I}(\bigcap_{i\in I} B_i \to C_i) \leqslant_{\Sigma} \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$. We can choose $B = \bigcap_{i\in I} B_i$ and conclude $\Gamma \vdash^{\Sigma} M : B \to A$, since $\bigcap_{i\in I} C_i \leqslant_{\Sigma} A$.

(3) By the v-soundness of Σ, we cannot have $v \sim_{\Sigma} B \to C$. Let I, B_i, C_i be as in Theorem 21(2), where $A \equiv B \to C$. Then $\bigcap_{i\in I}(B_i \to C_i) \leqslant_{\Sigma} B \to C$ implies that $\bigcap_{i\in J} C_i \leqslant_{\Sigma} C$, where $J = \{i \in I \mid B \leqslant_{\Sigma} B_i\}$, since Σ is beta. From $\Gamma, x{:}B_i \vdash^{\Sigma} M : C_i$ we can derive $\Gamma, x{:}B \vdash^{\Sigma} M : C_i$ using rule (⩽L), so by (∩I) we have $\Gamma, x{:}B \vdash^{\Sigma} M : \bigcap_{i\in J} C_i$. Finally, applying rule (⩽), we can conclude $\Gamma, x{:}B \vdash^{\Sigma} M : C$. □

4. Characterisation of Subject Reduction and Expansion

In the literature, to which we have provided many references in the previous sections, many models for the λ-calculus and a number of its restrictions have been shown to be finitary representable by means of (intersection) types. We now address, from the "intersection-type point-of-view", the generic requirements (a) and (b) concerning soundness discussed in the Introduction. In particular, we shall characterise those intersection-type systems in which types are preserved under various notions of conversions: β, η, together with some of their *restrictions* inspired by λ-calculi considered in the literature.

Let us first give the definitions of these restricted redexes.

Definition 23 (*Restricted redexes*).
(1) A redex $(\lambda x.M)N$ is a $\mathsf{var}\beta$-*redex* if N is a variable.
(2) A redex $(\lambda x.M)N$ is a $\mathsf{fun}\beta$-*redex* if N is an abstraction.

(3) A redex $(\lambda x.M)N$ is an $\mathsf{id}\beta$-*redex* if $x \in FV(M)$.
(4) A redex $(\lambda x.M)N$ is a $\mathsf{norm}\beta$-*redex* if N is a closed strongly normalising term.

The "call-by-value" λ-calculus is obtained by restricting to $\mathsf{var}\beta$- and $\mathsf{fun}\beta$-redexes (usually called β_v-redexes) [26], the $\lambda\mathbf{I}$-calculus by allowing to abstract only variables which occur free in the bodies (in this way we only get a proper subset of the set of $\mathsf{id}\beta$-redexes, whose elements are usually called $\beta\mathbf{I}$-redexes [4]) [13] and the $\lambda\mathbf{K}\mathsf{N}$-calculus by restricting to $\mathsf{var}\beta$-, $\mathsf{id}\beta$- and $\mathsf{norm}\beta$-redexes [23].

We shall deal now with rules of the form

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

where $\to_{\mathsf{R}}$ denotes the reduction relation obtained by restricting the contraction to the set of R-redexes. Admissibility of the above rules in a type assignment is usually referred to as *subject expansion* and *subject reduction*, respectively.

Theorem 24 (*Characterisation of subject* R-*conversion*).
(1) *If* $\Gamma \vdash^{\Sigma} M[x := N] : A$ *and* $\Gamma \vdash^{\Sigma} N : B$, *then* $\Gamma, x : B' \vdash^{\Sigma} M : A$ *and* $\Gamma \vdash^{\Sigma} N : B'$ *for some* $B' \in \mathbb{T}(\mathbb{C}^{\Sigma})$.
(2) (R-*expansion*) *Rule* (R-*exp*) *is admissible in* $\lambda\cap^{\Sigma}$ *iff for all* R-*redexes* $(\lambda x.M)N$ *and for all contexts* Γ:

N is typable in Γ *whenever* $M[x := N]$ *is typable in* Γ.

(3) (R-*reduction*) *Rule* (R-*red*) *is admissible in* $\lambda\cap^{\Sigma}$ *iff rule* $(\to \mathrm{I})$ *can be reversed for* R-*redexes, i.e. for all* Γ, M, A, B *such that* $(\lambda x.M)N$ *is a* R-*redex for some* N:

$$\Gamma \vdash^{\Sigma} \lambda x.M : B \to A \;\; \Rightarrow \;\; \Gamma, x{:}B \vdash^{\Sigma} M : A.$$

Proof. (1) If $A \sim_{\Sigma} \Omega$ we can choose $B' = \Omega$. Otherwise, the proof is by structural induction on M.

If $M \equiv y \neq x$ we can choose $B' = B$.

If $M \equiv x$, then $M[x := N] \equiv N$ and we can choose $B' = A$.

If $M \equiv M_1 M_2$ then, by Theorem 21(1), there are I, C_i, D_i such that $\Gamma \vdash^{\Sigma} M_1[x := N] : C_i \to D_i$, $\Gamma \vdash^{\Sigma} M_2[x := N] : C_i$, for all $i \in I$, and $\bigcap_{i\in I} D_i \leqslant_{\Sigma} A$. By induction, there are $B_i^{(1)}$, $B_i^{(2)}$ such that $\Gamma, x : B_i^{(1)} \vdash^{\Sigma} M_1 : C_i \to D_i$, $\Gamma \vdash^{\Sigma} N : B_i^{(1)}$, $\Gamma, x : B_i^{(2)} \vdash^{\Sigma} M_2 : C_i$ and $\Gamma \vdash^{\Sigma} N : B_i^{(2)}$ for $i \in I$. Then we can choose $B' = \bigcap_{i\in I}(B_i^{(1)} \cap B_i^{(2)})$. By rule $(\leqslant \mathrm{L})$, we get $\Gamma, x : B' \vdash^{\Sigma} M_1 : C_i \to D_i$, $\Gamma, x : B' \vdash^{\Sigma} M_2 : C_i$ and, by rule $(\cap \mathrm{I})$, $\Gamma \vdash^{\Sigma} N : B'$. So, we conclude using rules $(\to \mathrm{E})$, $(\cap \mathrm{I})$, and $(\leqslant)$.

If $M \equiv \lambda x.M'$ and $\nu \leqslant_{\Sigma} A$ then we can choose $B' = B$ since by rule (ν) $\Gamma, x : B \vdash^{\Sigma} \lambda x.M' : \nu$ and we conclude using rule $(\leqslant)$. Otherwise, by Theorem 21(2), there are I, C_i, D_i such that $\Gamma, y : C_i \vdash^{\Sigma} M'[x := N] : D_i$, for all $i \in I$ and $\bigcap_{i\in I}(C_i \to D_i) \leqslant_{\Sigma} A$. By induction, there are B_i such that $\Gamma, y : C_i, x : B_i \vdash^{\Sigma} M' : D_i$ and $\Gamma \vdash^{\Sigma} N : B_i$, for $i \in I$. Choosing $B' = \bigcap_{i\in I} B_i$, we get, by rule $(\leqslant \mathrm{L})$, $\Gamma, y : C_i, x : B' \vdash^{\Sigma} M' : D_i$, and, by rule $(\cap \mathrm{I})$, $\Gamma \vdash^{\Sigma} N : B'$. We conclude using rules $(\to \mathrm{I})$, $(\cap \mathrm{I})$ and $(\leqslant)$.

(2) $(\Rightarrow)$ Clearly, if N is not typable in the context Γ then also $(\lambda x.M)N$ has no type in Γ by Theorem 21(1).

$(\Leftarrow)$ It suffices to show that $\Gamma \vdash^{\Sigma} M[x := N] : A$ implies $\Gamma \vdash^{\Sigma} (\lambda x.M)N : A$ whenever $(\lambda x.M)N$ is an R-redex. By hypothesis, $\Gamma \vdash^{\Sigma} N : B$ for some B and then, by point (1) $\Gamma, x : B' \vdash^{\Sigma} M : A$ and $\Gamma \vdash^{\Sigma} N : B'$ for some B'. We conclude using rules $(\to \mathrm{I})$ and $(\to \mathrm{E})$.

(3) $(\Rightarrow)$ Assume $\Gamma \vdash^{\Sigma} \lambda x.M : B \to A$, which implies $\Gamma, y{:}B \vdash^{\Sigma} (\lambda x.M)y : A$ by rules (W) and $(\to \mathrm{E})$, for a fresh y. The admissibility of rule (R-*red*) gives us $\Gamma, y{:}B \vdash^{\Sigma} M[x := y] : A$. Hence, $\Gamma, x{:}B \vdash^{\Sigma} M : A$.

$(\Leftarrow)$ It suffices to show that $\Gamma \vdash^{\Sigma} (\lambda x.M)N : A$ implies $\Gamma \vdash^{\Sigma} M[x := N] : A$ whenever $(\lambda x.M)N$ is an R-redex. The case $A \sim_{\Sigma} \Omega$ is trivial for $\lambda\cap^{\Sigma}_{\Omega}$. Otherwise, by Theorem 21(1), there exist a finite set I and types B_i, C_i such that $\Gamma \vdash^{\Sigma} \lambda x.M : B_i \to C_i$, $\Gamma \vdash^{\Sigma} N : B_i$ and $\bigcap_{i\in I} C_i \leqslant_{\Sigma} A$. By hypothesis, we get $\Gamma, x{:}B_i \vdash^{\Sigma} M : C_i$. Then $\Gamma \vdash^{\Sigma} M[x := N] : C_i$ follows by an application of rule (C), and so we can conclude $\Gamma \vdash^{\Sigma} M[x := N] : A$ using rules $(\cap\mathrm{I})$ and $(\leqslant)$. □

By Theorems 22(3) and 24(3) we immediately get a condition which assures Subject β-reduction.

[4] For example, $(\lambda yx.y)z$ is an $\mathsf{id}\beta$-redex which is not a $\beta\mathbf{I}$-redex.

Corollary 25. *If Σ is ν-sound and beta then rule (β-red) is admissible in $\lambda\cap^{\Sigma}$.*

The condition of Theorem 24(2) above is immediately met in $\lambda\cap^{\Sigma}_{\Omega}$, in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ when $x\in FV(M)$ and in $\lambda\cap^{\Sigma}_{\nu}$ when N is an abstraction. We can discuss the admissibility and non-admissibility of restricted β-expansions for our type systems.

Corollary 26.
(1) *Rule* (varβ-*exp*) *is admissible in all $\lambda\cap^{\Sigma}_{\Omega}$, but never in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\nu}$.*
(2) *Rule* (funβ-*exp*) *is admissible in all $\lambda\cap^{\Sigma}_{\Omega}$ and $\lambda\cap^{\Sigma}_{\nu}$, but never in $\lambda\cap^{\Sigma}_{\mathcal{B}}$.*
(3) *Rule* (idβ-*exp*) *is admissible in all $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\Omega}$, but never in $\lambda\cap^{\Sigma}_{\nu}$.*
(4) *Rule* (normβ-*exp*) *is admissible in all $\lambda\cap^{\Sigma}$.*
(5) *Rule* (β-*exp*) *is admissible in all $\lambda\cap^{\Sigma}_{\Omega}$, but never in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\nu}$.*

Proof. Each of the five admissibilities but (4) follows from Theorem 24(2).

Item (4) is a consequence of Theorem 18, stating that each strongly normalising term is typable in all intersection-type systems from a suitable basis. So, all closed strongly normalising terms are typable in all intersection-type systems starting from the empty basis.

For the non-admissibility of rules (varβ-*exp*) and (β-*exp*) in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and in $\lambda\cap^{\Sigma}_{\nu}$, note that we can always derive $\vdash^{\Sigma}\lambda x.x : A\to A$, but by the Generation Lemmata I and II (Theorems 21(1) and 22(1)) we cannot derive the same type for $(\lambda yx.x)z$ from the empty basis without using $(Ax\text{-}\Omega)$.

An example showing that (funβ-*exp*) is not admissible in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ is $\vdash^{\Sigma}_{\mathcal{B}}\lambda x.x : A\to A$ and $\not\vdash^{\Sigma}_{\mathcal{B}}(\lambda yx.x)(\lambda t.z) : A\to A$.

An example showing that (idβ-*exp*) is not admissible in $\lambda\cap^{\Sigma}_{\nu}$ is $\vdash^{\Sigma}_{\nu}\lambda x.z : \nu$ and $\not\vdash^{\Sigma}_{\nu}(\lambda yx.y)z : \nu$. Rule (id$\beta$-*exp*) is not admissible also for terms of the $\lambda\mathbf{I}$-calculus since $\vdash^{\Sigma}_{\nu}\lambda x.zx : \nu$ and $\not\vdash^{\Sigma}_{\nu}(\lambda yx.yx)z : \nu$. □

Note that there are β-redexes that, without being normβ-redexes, are typable whenever their contracta are. As an example take $(\lambda x.y)y$.

We end this section with the characterisation of Subject Reduction and Expansion for the η-rule.

Theorem 27 (*Characterisation of subject η-conversion*).
(1) *Rule* (η-*exp*) *is admissible in $\lambda\cap^{\Sigma}$ iff Σ is eta.*
(2) *Rule* (η-*red*) *is admissible in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ iff Σ validates $\mathcal{CDV}$, in $\lambda\cap^{\Sigma}_{\Omega}$ iff Σ validates $\mathcal{BCD}$, and it is never admissible in $\lambda\cap^{\Sigma}_{\nu}$.*

Proof. (1) ($\Rightarrow$) Let $\diamondsuit\in\mathbb{C}^{\Sigma}$ be a constant that does not satisfy the first condition in Definition 9, i.e. $\nu\not\leqslant_{\Sigma}\diamondsuit$.

We can derive $x{:}\diamondsuit\vdash^{\Sigma} x : \diamondsuit$. To derive $x{:}\diamondsuit\vdash^{\Sigma}\lambda y.xy : \diamondsuit$ by Theorem 21(2) we need I, A_i, B_i such that $x{:}\diamondsuit$, $y{:}A_i\vdash^{\Sigma} xy : B_i$ for all $i\in I$ and $\bigcap_{i\in I}(A_i\to B_i)\leqslant_{\Sigma}\diamondsuit$. Let $I'=\{i\in I \mid B_i\not\sim_{\Sigma}\Omega\}$. For any $i\in I'$, by Theorem 21(1) we get $x{:}\diamondsuit$, $y{:}A_i\vdash^{\Sigma} x : D_{i,j}\to E_{i,j}$, $x{:}\diamondsuit$, $y{:}A_i\vdash^{\Sigma} y : D_{i,j}$, and $\bigcap_{j\in J_i}E_{i,j}\leqslant_{\Sigma}B_i$ for some $J_i, D_{i,j}, E_{i,j}$. By Theorem 22(1) we have $\diamondsuit\leqslant_{\Sigma}D_{i,j}\to E_{i,j}$ and $A_i\leqslant_{\Sigma}D_{i,j}$ for all $i\in I'$ and $j\in J_i$. So we conclude

$$\bigcap_{i\in I}(A_i\to B_i)\leqslant_{\Sigma}\diamondsuit\leqslant_{\Sigma}\bigcap_{i\in I'}\left(\bigcap_{j\in J_i}(D_{i,j}\to E_{i,j})\right)\quad\forall i\in I'.\ A_i\leqslant_{\Sigma}\bigcap_{j\in J_i}D_{i,j}\ \ \&\ \ \bigcap_{j\in J_i}E_{i,j}\leqslant_{\Sigma}B_i.$$

($\Leftarrow$) The proof that $\Gamma\vdash^{\Sigma}M : A$ implies $\Gamma\vdash^{\Sigma}\lambda x.Mx : A$, where x is fresh, is by induction on the structure of A. The unique non-trivial case is when $A\equiv\psi$ is a type constant not greater than ν. In this case, we use the fact that Σ is eta in order to do the derivation discussed in the proof of ($\Rightarrow$). In details, suppose that $\Gamma\vdash^{\Sigma}M : \psi$ for some $\psi\in\mathbb{C}^{\Sigma}$ such that $\nu\not\leqslant_{\Sigma}\psi$ and moreover:

$$\bigcap_{i\in I}(A_i\to B_i)\leqslant_{\Sigma}\psi\leqslant_{\Sigma}\bigcap_{i\in I'}\left(\bigcap_{j\in J_i}(D_{i,j}\to E_{i,j})\right)\quad\forall i\in I'.\ A_i\leqslant_{\Sigma}\bigcap_{j\in J_i}D_{i,j}\ \ \&\ \ \bigcap_{j\in J_i}E_{i,j}\leqslant_{\Sigma}B_i,$$

where $I'=\{i\in I \mid B_i\not\sim_{\Sigma}\Omega\}$. By rule ($\leqslant$), we can derive $\Gamma\vdash^{\Sigma}M : D_{i,j}\to E_{i,j}$ for all $i\in I'$, $j\in J_i$, and so $\Gamma, x{:}D_{i,j}\vdash^{\Sigma}Mx : E_{i,j}$ by rule ($\to$E). From rules ($\leqslant$ L), ($\cap$I) and ($\leqslant$) we get $\Gamma, x{:}A_i\vdash^{\Sigma}Mx : B_i$ for all $i\in I'$. Consider

now $i \in I \setminus I'$. In such a case, since $B_i \sim_\Sigma \Omega$, we get immediately, using axiom (Ax-Ω) and rule ($\leqslant$), $\Gamma, x{:}A_i \vdash^\Sigma Mx : B_i$. Therefore, for any $i \in I$, we get $\Gamma \vdash^\Sigma \lambda x.Mx : A_i \to B_i$ using rule ($\to$I). So we can conclude by ($\cap$I) and ($\leqslant$) that $\Gamma \vdash^\Sigma \lambda x.Mx : \psi$.

(2) ($\Rightarrow$) Let us assume that Σ does not validate axiom ($\to$-$\cap$), i.e. that there are types A, B, C such that $(A \to B) \cap (A \to C) \not\leqslant_\Sigma A \to B \cap C$. We can derive $x{:}(A \to B) \cap (A \to C) \vdash^\Sigma_{\mathcal{B}} \lambda y.xy : A \to B \cap C$ using rules ($\leqslant$), ($\to$E), ($\cap$I), and ($\to$I), but $x : A \to B \cap C$ cannot be derived from $x{:}(A \to B) \cap (A \to C)$ by Theorem 22(1). Now, suppose that Σ does not validate rule (η), i.e. that there are types A, B, C, D such that $A \leqslant_\Sigma B$ and $C \leqslant_\Sigma D$ but $B \to C \not\leqslant_\Sigma A \to D$. We can derive $x{:}B \to C \vdash^\Sigma_{\mathcal{B}} \lambda y.xy : A \to D$ using rules ($\leqslant$), ($\to$E), and ($\to$I), but $x{:}B \to C \not\vdash^\Sigma_{\mathcal{B}} x : A \to D$ by Theorem 22(1).

If $\Omega \in \mathbb{C}^\Sigma$ we get $x{:}\Omega \vdash^\Sigma_\Omega \lambda y.xy : \Omega \to \Omega$ by axiom (Ax-Ω) and rule ($\to$I). By Theorem 22(1), we can derive $x{:}\Omega \vdash^\Sigma_\Omega x : \Omega \to \Omega$ iff $\Omega \leqslant_\Sigma \Omega \to \Omega$, i.e. iff Σ validates axiom (Ω-η).

If $\nu \in \mathbb{C}^\Sigma$ we get $\vdash^\Sigma_\nu \lambda y.xy : \nu$ by axiom (Ax-ν), but we cannot derive $x : \nu$ from the empty basis by Theorem 22(1).

($\Leftarrow$) We prove that under the given conditions on type preorders $\Gamma \vdash^\Sigma \lambda x.Mx : A$ and $x \notin FV(M)$ imply $\Gamma \vdash^\Sigma M : A$. We give the proof for $\lambda\cap^\Sigma_\Omega$, that one for $\lambda\cap^\Sigma_{\mathcal{B}}$ being similar and simpler. By Theorem 21(2), $\Gamma \vdash^\Sigma_\Omega \lambda x.Mx : A$ implies that there are I, B_i, C_i such that $\Gamma, x{:}B_i \vdash^\Sigma_\Omega Mx : C_i$ and $\bigcap_{i\in I}(B_i \to C_i) \leqslant_\Sigma A$. If for some i we get $C_i \sim_\Sigma \Omega$, then we can obtain $B_i \to C_i \sim_\Sigma \Omega$ by axiom (Ω-η) and rule (η). Therefore, we can forget those $B_i \to C_i$. Otherwise, $\Gamma, x{:}B_i \vdash^\Sigma_\Omega Mx : C_i$ implies by Theorem 22(2) and rule (S) that $\Gamma \vdash^\Sigma_\Omega M : D_i \to C_i$, and $\Gamma, x{:}B_i \vdash^\Sigma_\Omega x : D_i$, for some D_i. By Theorem 22(1) we get $B_i \leqslant_\Sigma D_i$, so we can derive $\Gamma \vdash^\Sigma_\Omega M : B_i \to C_i$ using rule ($\leqslant$), since $D_i \to C_i \leqslant_\Sigma B_i \to C_i$ by rule (η). Rule ($\cap$I) implies $\Gamma \vdash^\Sigma_\Omega M : \bigcap_{i\in I}(B_i \to C_i)$. So we can conclude $\Gamma \vdash^\Sigma_\Omega M : A$ using rule ($\leqslant$). □

5. Filter λ-structures and filter models

In this section, we shall see how the results obtained in the previous sections can be used to prove characterisation results concerning domains defined by means of intersection types, the so-called filter λ-structures. In particular, necessary and sufficient conditions will be given that characterise those filter λ-structures that are also models for the (restricted) λ-calculi.

Let us begin with a short discussion about how it is possible to interpret types. There are essentially *two* semantics for intersection types.

One is the *set-theoretical* semantics, originally introduced in [9], generalising the one given by Scott for simple types. The meanings of types are subsets of the domain of discourse, arrow types are defined as *logical predicates* and intersection is set-theoretic intersection.

The second semantics, which arises in the wake of Stone Duality results (see [1,12,31]), views types as *compact elements* of Plotkin's λ-structures [27]. According to this interpretation, the type Ω denotes the least element, intersections denote joins of compact elements, and arrow types allow to internalise the space of continuous endomorphisms. By duality, type preorders give rise to *filter λ-structures*, where the interpretation of λ-terms can be given through a finitary logical description.

In order to introduce filter λ-structures, let us give the appropriate notion of filter over a type preorder.

Definition 28 (Σ-*filters*). Let Σ be a type preorder.

(1) A Σ-filter (or a filter over $\mathbb{T}(\mathbb{C}^\Sigma)$) is a set $\Xi \subseteq \mathbb{T}(\mathbb{C}^\Sigma)$ such that
   (a) if $\Omega \in \mathbb{C}^\Sigma$ then $\Omega \in \Xi$;
   (b) if $A \leqslant_\Sigma B$ and $A \in \Xi$, then $B \in \Xi$;
   (c) if A, $B \in \Xi$, then $A \cap B \in \Xi$.

(2) $\mathcal{F}^\Sigma$ the set of Σ-filters over $\mathbb{T}(\mathbb{C}^\Sigma)$.

(3) If $\Xi \subseteq \mathbb{T}(\mathbb{C}^\Sigma)$, $\uparrow\Xi$ denotes the Σ-filter generated by Ξ.

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

It is well known that $\mathcal{F}^\Sigma$ is an ω-algebraic lattice, whose poset of compact (or finite) elements is isomorphic to the reversed poset obtained by quotienting the preorder on $\mathbb{T}(\mathbb{C}^\Sigma)$ by $\sim_\Sigma$. Which means that compact elements are the filters

of the form $\uparrow A$ for some type A, the top element is $\mathbb{T}(\mathbb{C}^\Sigma)$, and the bottom element is $\uparrow\Omega$ when $\Omega\in\mathbb{C}^\Sigma$ and $\emptyset$ otherwise. Moreover, the join of two filters is the filter induced by their union and the meet of two filters is their intersection, i.e.

$$\Xi \sqcup \Upsilon = \uparrow(\Xi \cup \Upsilon),$$
$$\Xi \sqcap \Upsilon = \Xi \cap \Upsilon.$$

We now turn the space of filters into an applicative structure.

Definition 29 (*Application*). Application $\_ \cdot \_ : \mathcal{F}^\Sigma \times \mathcal{F}^\Sigma \to \mathcal{F}^\Sigma$ is defined as

$$\Xi \cdot \Upsilon = \uparrow\{B \mid \exists A\in\Upsilon. A \to B\in\Xi\}.$$

Taking the Stone duality view-point, the interpretation of terms coincides with the sets of types which are deducible for them.

Definition 30. For any λ-term M and environment $\rho : Var \to \mathcal{F}^\Sigma \setminus \{\emptyset\}$,

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

where Var is the set of term variables and $\Gamma \vDash \rho$ if and only if $(x : B)\in\Gamma$ implies $B\in\rho(x)$.

We call *filter λ-structure* the triple $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma\rangle$.

By rules (Ω), $(\leqslant)$ and $(\cap\mathrm{I})$, the interpretations of all λ-terms are filters.

Dropping the empty set from the codomain of environments is necessary for obtaining models. First of all, note that the empty set is a filter only if $\Omega \notin \mathbb{C}^\Sigma$. Clearly, any reasonable interpretation of λ-terms must give the same meaning to the terms z and $(\lambda y.z)x$. If we would allow $\rho(z) = \Xi \neq \emptyset$ and $\rho(x) = \emptyset$ we would get $[\![z]\!]^\Sigma_\rho = \Xi$ and $[\![(\lambda y.z)x]\!]^\Sigma_\rho = \emptyset$: in fact no type is derivable for x from a basis which does not contain x when $\Omega \notin \mathbb{C}^\Sigma$. This example is obviously related to the fact that rule (idβ-*exp*) is admissible only when $\Omega\in\mathbb{C}^\Sigma$.

Remark 31. In the literature (see for instance [16]), filter λ-structures are often referred to as triples $\langle \mathcal{F}^\Sigma, \mathbb{F}^\Sigma, \mathbb{G}^\Sigma\rangle$ where the maps $\mathbb{F}^\Sigma : \mathcal{F}^\Sigma \to [\mathcal{F}^\Sigma \to \mathcal{F}^\Sigma]$ and $\mathbb{G}^\Sigma : [\mathcal{F}^\Sigma \to \mathcal{F}^\Sigma] \to \mathcal{F}^\Sigma$ are defined by

$$\mathbb{F}^\Sigma(\Xi) = \lambda\Upsilon\in\mathcal{F}^\Sigma.\Xi \cdot \Upsilon;$$
$$\mathbb{G}^\Sigma(f) = \begin{cases} \uparrow\{A \to B \mid B\in f(\uparrow A)\} \sqcup \uparrow v & \text{if } v\in\mathbb{C}, \\ \uparrow\{A \to B \mid B\in f(\uparrow A)\} & \text{otherwise.} \end{cases}$$

Actually our definition of filter λ-structure coincides with this last one, since application "$\cdot$" allows to recover both $\mathbb{F}^\Sigma$ and $\mathbb{G}^\Sigma$. Moreover, the interpretation $[\![\ ]\!]^\Sigma$ coincides with the interpretation of λ-terms induced in the standard way by $\mathbb{F}^\Sigma$ and $\mathbb{G}^\Sigma$. We prefer here the definition of filter λ-structures as triples $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma\rangle$ since it is closer to the syntactic perspective of the previous sections.

The notion of restricted redexes introduced in Definition 23 leads us to consider correspondingly notions of restricted λ-models: first, we adapt the classical definition of λ-model à la Hindley–Longo [21] to encompass the various notions of reduction, then we characterise the filter structures which induce these models. To accommodate "call-by-value" λ-calculus we allow the codomains of environments to be not necessarily the whole interpretation domains, but suitable subsets of them.

Definition 32 (λ-*models*). A *model* for the (restricted) λ-calculus λ_{R} (i.e. of the calculus whose redexes are exactly the R-redexes) consists of a triple $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}}\rangle$ such that $\mathcal{D}$ is a set, $\cdot : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$, $Env : Var \to \mathcal{V}$ for some $\mathcal{V} \subseteq \mathcal{D}$ and the interpretation function $[\![\ ]\!]^{\mathcal{D}} : \Lambda \times Env \to \mathcal{D}$ satisfies:

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

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

(3) $[\![\lambda x.M]\!]^{\mathcal{D}}_\rho \cdot [\![N]\!]^{\mathcal{D}}_\rho = [\![M]\!]^{\mathcal{D}}_{\rho[x:=[\![N]\!]^{\mathcal{D}}_\rho]}$ for all R-redexes $(\lambda x.M)N$;

(4) If $\rho(x) = \rho'(x)$ for all $x \in \mathrm{FV}(M)$, then $[\![M]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho'}$;
(5) If $y \notin \mathrm{FV}(M)$, then $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda y.M[x := y]]\!]^{\mathcal{D}}_{\rho}$;
(6) If $\forall d \in \mathcal{D}.[\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} = [\![N]\!]^{\mathcal{D}}_{\rho[x:=d]}$, then $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\rho}$.
The restricted model $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ is *extensional* if moreover when $x \notin \mathrm{FV}(M)$:

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

Actually, using the Generation Lemmata, we can prove that all filter λ-structures satisfy all the points of the previous definition but the third one.

A direct counterexample to the third point is easy. Considering R as the set of β-redexes, take for instance the preorder $\Sigma^{\dagger}$ defined by $\mathbb{C}^{\dagger} = \{\Omega, \varphi\}$, and $\dagger = \{(\dagger)\}$, where

$$(\dagger) \quad \Omega \to \Omega \sim \Omega \to \varphi.$$

Because of $(\dagger)$ and $(\leqslant)$, we have $\vdash^{\dagger}_{\Omega} \lambda x.x : \Omega \to \varphi$. Let $\rho(y) = \uparrow\Omega$. Then, by $(\to \mathrm{E})$, we get $\vdash^{\dagger}_{\Omega} (\lambda x.x)y : \varphi$. So

$$\begin{array}{rcll} [\![(\lambda x.x)y]\!]^{\dagger}_{\rho} & = & \{A \mid \exists \Gamma \vDash \rho.\ \Gamma \vdash^{\dagger}_{\Omega} (\lambda x.x)y : A\} & \text{by definition of interpretation} \\ & \supseteq & \uparrow\varphi & \text{by above} \\ & \neq & \uparrow\Omega & \text{since } \varphi \notin \uparrow\Omega \\ & = & [\![y]\!]^{\dagger}_{\rho}. & \end{array}$$

Lemma 33. *For all type preorders Σ the interpretation function $[\![\ ]\!]^{\Sigma}$ satisfies conditions* (1), (2), (4)–(6) *of Definition* 32.

Proof. (1) Follows immediately from Definition 30 and Theorem 22(1).

(2) Let $A \in [\![MN]\!]^{\Sigma}_{\rho}$. The case $A \sim_{\Sigma} \Omega$ is trivial. Otherwise, there exists a Σ-basis Γ such that $\Gamma \vDash \rho$ and $\Gamma \vdash^{\Sigma} MN : A$. By Theorem 21(1), there exist I, and $B_i, C_i \in \mathbb{T}(\mathbb{C}^{\Sigma})$ such that $\Gamma \vdash^{\Sigma} M : B_i \to C_i$, $\Gamma \vdash^{\Sigma} N : B_i$ for all $i \in I$, and $\bigcap_{i\in I} C_i \leqslant_{\Sigma} A$. Hence $B_i \in [\![N]\!]^{\Sigma}_{\rho}$ and $B_i \to C_i \in [\![M]\!]^{\Sigma}_{\rho}$, for all $i \in I$. By definition of application, it follows $C_i \in [\![M]\!]^{\Sigma}_{\rho} \cdot [\![N]\!]^{\Sigma}_{\rho}$ for all $i \in I$, and this implies $A \in [\![M]\!]^{\Sigma}_{\rho} \cdot [\![N]\!]^{\Sigma}_{\rho}$, being $[\![M]\!]^{\Sigma}_{\rho} \cdot [\![N]\!]^{\Sigma}_{\rho}$ a filter.

Let now $A \in [\![M]\!]^{\Sigma}_{\rho} \cdot [\![N]\!]^{\Sigma}_{\rho}$. Then there exist I, and $B_i, C_i \in \mathbb{T}(\mathbb{C}^{\Sigma})$ such that $B_i \to C_i \in [\![M]\!]^{\Sigma}_{\rho}$, $B_i \in [\![N]\!]^{\Sigma}_{\rho}$, for all $i \in I$, and $\bigcap_{i\in I} C_i \leqslant_{\Sigma} A$. Hence there exist Σ-bases, Γ_i and Γ'_i, such that $\Gamma_i \vDash \rho$, $\Gamma'_i \vDash \rho$, and moreover $\Gamma_i \vdash^{\Sigma} M : B_i \to C_i$, $\Gamma'_i \vdash^{\Sigma} N : B_i$. Consider the Σ-basis $\Gamma'' = \biguplus_{i\in I}(\Gamma_i \uplus \Gamma'_i)$. We have $\Gamma'' \vDash \rho$, $\Gamma'' \vdash^{\Sigma} M : B_i \to C_i$ and $\Gamma'' \vdash^{\Sigma} N : B_i$. Using rules $(\to \mathrm{E})$, $(\cap \mathrm{I})$ and $(\leqslant)$ we deduce $\Gamma'' \vdash^{\Sigma} MN : A$, so we conclude $A \in [\![MN]\!]^{\Sigma}_{\rho}$.

(4) and (5) are trivial.

(6) Suppose that the premise holds and $A \in [\![\lambda x.M]\!]^{\Sigma}_{\rho}$. The case $\nu \leqslant_{\Sigma} A$ is trivial. Otherwise, there is a Σ-basis Γ such that $\Gamma \vDash \rho$ and $\Gamma \vdash^{\Sigma} \lambda x.M : A$. Since $x \notin \mathrm{FV}(\lambda x.M)$, by rule (S) we can assume $x \notin \Gamma$. By Theorem 21(2), there exist I and $B_i, C_i \in \mathbb{T}(\mathbb{C}^{\Sigma})$ such that $\Gamma, x : B_i \vdash^{\Sigma} M : C_i$ for all $i \in I$. Then $C_i \in [\![M]\!]^{\Sigma}_{\rho[x:=\uparrow B_i]}$: by the premise this implies $C_i \in [\![N]\!]^{\Sigma}_{\rho[x:=\uparrow B_i]}$, and so $\Gamma_i, x : B_i \vdash^{\Sigma} N : C_i$ for some Σ-basis Γ_i such that $\Gamma_i \vDash \rho$. Choosing $\Gamma' = \biguplus_{i\in I} \Gamma_i$, we have $\Gamma' \vDash \rho$ and $\Gamma', x : B_i \vdash^{\Sigma} N : C_i$ for all $i \in I$. Using rules $(\to \mathrm{I})$, $(\cap \mathrm{I})$ and $(\leqslant)$ we deduce $\Gamma' \vdash^{\Sigma} \lambda x.N : A$, so we conclude $[\![\lambda x.M]\!]^{\Sigma}_{\rho} \subseteq [\![\lambda x.N]\!]^{\Sigma}_{\rho}$. Similarly, one can prove $[\![\lambda x.N]\!]^{\Sigma}_{\rho} \subseteq [\![\lambda x.M]\!]^{\Sigma}_{\rho}$. □

Due to the previous lemma, a filter λ-structure is a *model* (i.e. a *filter model*) *of the λ-calculus* λ_{R} iff the interpretation function $[\![\ ]\!]^{\Sigma}$ equates the R-redexes with their contracta, that is it satisfies the condition (3) of Definition 32:

$$[\![(\lambda x.M)N]\!]^{\Sigma}_{\rho} = [\![M]\!]^{\Sigma}_{\rho[x:=[\![N]\!]^{\Sigma}_{\rho}]} \quad \text{for all } \mathsf{R}\text{-redexes } (\lambda x.M)N.$$

For the successive development it is handy to split the above condition in the following two conditions on type assignment systems which are similar to the rules (R-*exp*) and (R-*red*).

Definition 34.
(1) Condition (R-$[\![exp]\!]$): for all x, M, N, Γ, ρ, if $(\lambda x.M)N$ is an R-redex, $\Gamma \vdash^{\Sigma} M[x := N] : A$, and $\Gamma \vDash \rho$, then there is a Σ-basis Γ' such that $\Gamma' \vDash \rho$ and $\Gamma' \vdash^{\Sigma} (\lambda x.M)N : A$.
(2) Condition (R-$[\![red]\!]$): for all x, M, N, Γ, ρ, if $(\lambda x.M)N$ is an R-redex, $\Gamma \vdash^{\Sigma} (\lambda x.M)N : A$, and $\Gamma \vDash \rho$, then there is a Σ-basis Γ' such that $\Gamma' \vDash \rho$ and $\Gamma' \vdash^{\Sigma} M[x := N] : A$.

Clearly, these conditions are more permissive than the corresponding rules: i.e. the admissibility of the rule implies the validity of the condition, but not vice versa.

Theorem 35.
(1) *Condition* (varβ-$[\![exp]\!]$) *holds in all* $\lambda\cap^{\Sigma}$.
(2) *Condition* (funβ-$[\![exp]\!]$) *holds in all* $\lambda\cap^{\Sigma}_{\Omega}$ *and* $\lambda\cap^{\Sigma}_{\nu}$, *and in* $\lambda\cap^{\Sigma}_{\mathcal{B}}$ *proviso that for all* ρ, x, M *there are a* Σ*-basis* Γ *and a type* $A \in \mathbb{T}(\mathbb{C}^{\Sigma})$ *such that* $\Gamma \vDash \rho$ *and* $\Gamma \vdash^{\Sigma}_{\mathcal{B}} \lambda x.M : A$.
(3) *Condition* (idβ-$[\![exp]\!]$) *holds in all* $\lambda\cap^{\Sigma}_{\mathcal{B}}$ *and* $\lambda\cap^{\Sigma}_{\Omega}$, *and in* $\lambda\cap^{\Sigma}_{\nu}$ *proviso that for all* ρ, M *there are a* Σ*-basis* Γ *and a type* $A \in \mathbb{T}(\mathbb{C}^{\Sigma})$ *such that* $\Gamma \vDash \rho$ *and* $\Gamma \vdash^{\Sigma}_{\nu} M : A$.
(4) *Condition* (normβ-$[\![exp]\!]$) *holds in all* $\lambda\cap^{\Sigma}$.
(5) *Condition* (β-$[\![exp]\!]$) *holds in all* $\lambda\cap^{\Sigma}_{\Omega}$, *and in* $\lambda\cap^{\Sigma}_{\mathcal{B}}$ *and* $\lambda\cap^{\Sigma}_{\nu}$ *proviso that for all* ρ, M *there are a* Σ*-basis* Γ *and a type* $A \in \mathbb{T}(\mathbb{C}^{\Sigma})$ *such that* $\Gamma \vDash \rho$ *and* $\Gamma \vdash^{\Sigma} M : A$.

Proof. Admissibility of rule (R-*exp*) implies validity of condition (R-$[\![exp]\!]$), hence by Corollary 26 we have that:
- Condition (funβ-$[\![exp]\!]$) holds in all $\lambda\cap^{\Sigma}_{\Omega}$ and $\lambda\cap^{\Sigma}_{\nu}$;
- Condition (idβ-$[\![exp]\!]$) holds in all $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\Omega}$;
- Condition (normβ-$[\![exp]\!]$) holds in all $\lambda\cap^{\Sigma}$;
- Condition (β-$[\![exp]\!]$) holds in all $\lambda\cap^{\Sigma}_{\Omega}$.

For the remaining cases, note that $\Gamma \vDash \rho$ and $\Gamma' \vDash \rho$ imply $\Gamma \uplus \Gamma' \vDash \rho$ by definition of $\vDash$. Moreover, if $\Gamma \vdash^{\Sigma} M : A$ we get $\Gamma \uplus \Gamma' \vdash^{\Sigma} M : A$ by rules ($\leqslant$ L) and (W) for all Γ'. Therefore, by Theorem 24(2), condition (R-$[\![exp]\!]$) can be rewritten as follows:

> for all x, M, N, Γ, ρ, if $(\lambda x.M)N$ is an R-redex, $\Gamma \vdash^{\Sigma} M[x := N] : A$, and $\Gamma \vDash \rho$, then there are a Σ-basis Γ' and a type $B \in \mathbb{T}(\mathbb{C}^{\Sigma})$ such that $\Gamma' \vDash \rho$ and $\Gamma' \vdash^{\Sigma} N : B$.

(1) We get $\{x : B\} \vdash^{\Sigma} x : B$ for all $B \in \rho(x)$: recall that by definition $\rho(x)$ is never empty for all ρ and x.
(2) The condition for $\lambda\cap^{\Sigma}_{\mathcal{B}}$ is clearly sufficient. It is also necessary: $(\lambda yz.z)(\lambda x.M)$ is a funβ-redex for all x, M and $\vdash^{\Sigma}_{\mathcal{B}} \lambda z.z : A \to A$.
(3) The condition for $\lambda\cap^{\Sigma}_{\nu}$ is clearly sufficient. It is also necessary: $(\lambda yz.y)M$ is an idβ-redex for all M and $\vdash^{\Sigma}_{\nu} \lambda z.M : \nu$.
(4) The condition for $\lambda\cap^{\Sigma}_{\mathcal{B}}$ and $\lambda\cap^{\Sigma}_{\nu}$ is clearly sufficient. It is also necessary since $(\lambda yz.z)M$ is a β-redex for all M and $\vdash^{\Sigma} \lambda z.z : A \to A$. □

If $(\lambda x.M)N$ being an R-redex implies that so is $(\lambda x.M)y$, for a fresh variable y, then condition (R-$[\![red]\!]$) is equivalent to:

> Condition (R-$[\![red]\!]$-$*$) : for all x, M, Γ, ρ, if $(\lambda x.M)N$ is an R-redex for some N, $\Gamma \vdash^{\Sigma} \lambda x.M : B \to A$, and $\Gamma \vDash \rho$, then there is a Σ-basis Γ' such that $\Gamma' \vDash \rho$ and $\Gamma', x{:}B \vdash^{\Sigma} M : A$.

as proved in the following theorem.

Theorem 36. *Conditions* (R-$[\![red]\!]$) *and* (R-$[\![red]\!]$-$*$) *are equivalent, proviso* $(\lambda x.M)N$ *being an* R*-redex implies that so is* $(\lambda x.M)y$, *for a fresh variable* y.

Proof. ($\Rightarrow$) Let $(\lambda x.M)N$ be an R-redex for some N. Assume $\Gamma \vdash^{\Sigma} \lambda x.M : B \to A$, which implies $\Gamma, y{:}B \vdash^{\Sigma} (\lambda x.M)y : A$ by rule ($\to$ E) for a fresh y. Note that by assumption $(\lambda x.M)y$ is a R-redex. Note also that $\Gamma \vDash \rho$ implies Γ,

$y{:}B \vDash \rho[y := \uparrow B]$. By condition (R-$[\![red]\!]$) there is $\Gamma' \vDash \rho[y := \uparrow B]$ such that $\Gamma' \vdash^{\Sigma} M[x := y] : A$. Let $\Gamma'' = \{z{:}C \in \Gamma' \mid z \neq y\}$: by construction $\Gamma'' \vDash \rho$ and by rule ($\leqslant$ L) Γ'', $y{:}B \vdash^{\Sigma} M[x := y] : A$. Hence, Γ'', $x{:}B \vdash^{\Sigma} M : A$.

($\Leftarrow$) Let $\Gamma \vdash^{\Sigma} (\lambda x.M)N : A$ and $(\lambda x.M)N$ be an R-redex. The case $A \sim_{\Sigma} \Omega$ is trivial for $\lambda\cap^{\Sigma}_{\Omega}$. Otherwise, by Theorem 21 (1), there exist a finite set I and types B_i, C_i such that $\Gamma \vdash^{\Sigma} \lambda x.M : B_i \to C_i$, $\Gamma \vdash^{\Sigma} N : B_i$ and $\bigcap_{i\in I} C_i \leqslant_{\Sigma} A$. By condition (R-$[\![red]\!]$-$*$) there are Γ'_i such that $\Gamma'_i \vDash \rho$ and Γ'_i, $x{:}B_i \vdash^{\Sigma} M : C_i$ for all $i \in I$. Let $\Gamma'' = \Gamma \uplus (\biguplus_{i\in I} \Gamma_i)$: we get $\Gamma'' \vDash \rho$ by definition of $\vDash$. We deduce by ($\leqslant$ L) and (W) Γ'', $x{:}B_i \vdash^{\Sigma} M : C_i$ and $\Gamma'' \vdash^{\Sigma} N : B_i$ for all $i \in I$. Then $\Gamma'' \vdash^{\Sigma} M[x := N] : C_i$ follows by an application of rule (C), and so we can conclude $\Gamma'' \vdash^{\Sigma} M[x := N] : A$ using rules ($\cap$I) and ($\leqslant$). □

Remark that the condition of previous theorem is satisfied when the set of R-redexes includes the set of varβ-redexes.

We end the section by giving the characterisations of the type preorders inducing models of (restricted) λ-calculi. These characterisations, which are generalisations of the corresponding result in [11], follow easily from Theorems 35 and 36.

Theorem 37 (*Characterisations of (restricted) filter models*). $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma}\rangle$ *is a model of the*

(1) "*call-by-value*" λ*-calculus iff for all* Γ, x, M, A, B, ρ:
   (a) *if* $\Gamma \vdash^{\Sigma} \lambda x.M : A \to B$ *and* $\Gamma \vDash \rho$ *then* Γ', $x : A \vdash^{\Sigma} M : B$ *and* $\Gamma' \vDash \rho$ *for some* Σ*-basis* Γ';
   (b) $[\![\lambda x.M]\!]^{\Sigma}_{\rho} \neq \emptyset$;

(2) $\lambda\mathbf{I}$*-calculus iff for all* Γ, x, M, A, B, ρ *such that* $x \in FV(M)$:
   (a) *if* $\Gamma \vdash^{\Sigma} \lambda x.M : A \to B$ *and* $\Gamma \vDash \rho$ *then* Γ', $x : A \vdash^{\Sigma} M : B$ *and* $\Gamma' \vDash \rho$ *for some* Σ*-basis* Γ';
   (b) *if* $v \in \mathbb{C}^{\Sigma}$ *then* $[\![M]\!]^{\Sigma}_{\rho} \neq \emptyset$;

(3) $\lambda\mathbf{K}\mathsf{N}$*-calculus iff for all* Γ, x, M, A, B, ρ:
   (a) *if* $\Gamma \vdash^{\Sigma} \lambda x.M : A \to B$ *and* $\Gamma \vDash \rho$ *then* Γ', $x : A \vdash^{\Sigma} M : B$ *and* $\Gamma' \vDash \rho$ *for some* Σ*-basis* Γ';
   (b) *if* $v \in \mathbb{C}^{\Sigma}$ *then* $[\![M]\!]^{\Sigma}_{\rho} \neq \emptyset$;

(4) *whole* λ*-calculus iff for all* Γ, x, M, A, B, ρ:
   (a) *if* $\Gamma \vdash^{\Sigma} \lambda x.M : A \to B$ *and* $\Gamma \vDash \rho$ *then* Γ', $x : A \vdash^{\Sigma} M : B$ *and* $\Gamma' \vDash \rho$ *for some* Σ*-basis* Γ';
   (b) $[\![M]\!]^{\Sigma}_{\rho} \neq \emptyset$.

Proof. First of all note that all considered calculi satisfy the condition of Theorem 36. More precisely in all these calculi but in the $\lambda\mathbf{I}$-calculus the set of redexes includes the set of varβ-redexes. Instead for all variables y we have that $(\lambda x.M)y$ is a $\beta\mathbf{I}$-redex whenever $(\lambda x.M)N$ is a $\beta\mathbf{I}$-redex.

Condition (a) specialises condition (R-$[\![red]\!]$-$*$) for the considered restrictions of λ-calculus, and therefore by Theorem 36, (a) is necessary and sufficient to assure:

$$[\![(\lambda x.M)N]\!]^{\Sigma}_{\rho} \subseteq [\![M]\!]^{\Sigma}_{\rho[x:=[\![N]\!]^{\Sigma}_{\rho}]}.$$

Taking into account that $[\![M]\!]^{\Sigma}_{\rho} \neq \emptyset$ iff there are $\Gamma \vDash \rho$, A such that $\Gamma \vdash^{\Sigma} M : A$, Theorem 35 implies that condition (b) is necessary and sufficient to assure:

$$[\![M]\!]^{\Sigma}_{\rho[x:=[\![N]\!]^{\Sigma}_{\rho}]} \subseteq [\![(\lambda x.M)N]\!]^{\Sigma}_{\rho}. \qquad \square$$

As an immediate consequence of Theorems 37(4) and 22(3), $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma}\rangle$ is a λ-model whenever Σ is a beta theory and $\Omega \in \mathbb{C}^{\Sigma}$.

We can also characterise filter models which are extensional using Theorem 27. Note that for the η-rule the possibility of changing basis (in agreement with a fixed environment) plays a role only if $v \in \mathbb{C}$, since in all other cases the subformula property holds and η-convertible terms have the same set of free variables.

Theorem 38 (*Characterisation of extensional (restricted) filter models*). *Let* Σ *be a type preorder. The filter* λ*-structure* $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma}\rangle$ *is an extensional filter model of the restricted* λ*-calculus* λ_{R} *iff it is a model of* λ_{R}, Σ *is an eta-type preorder which validates* $\mathcal{CDV}$, *and moreover if* $\Omega \in \mathbb{C}^{\Sigma}$ *then* Σ *validates axiom* (Ω-η), *if* $v \in \mathbb{C}^{\Sigma}$ *then* $v \in [\![M]\!]^{\Sigma}_{\rho}$ *for all* M, ρ.

Proof. ($\Rightarrow$) Let $\varphi\in\mathbb{C}^\Sigma$ be a constant that satisfies neither conditions of Definition 9. One can show that $\varphi\notin[\![\lambda y.xy]\!]^\Sigma_{\rho[x:=\uparrow\varphi]}$: this implies that Σ must be eta.

We have $A\to B\cap C\in[\![\lambda y.xy]\!]^\Sigma_{\rho[x:=\uparrow(A\to B)\cap(A\to C)]}$ for all A, B, C, but $A\to B\cap C\in\uparrow(A\to B)\cap(A\to C)$ only if Σ validates axiom ($\to$-$\cap$). Similarly, one can show that Σ must validate axiom (η) always, and axiom (Ω-η) when $\Omega\in\mathbb{C}^\Sigma$. Lastly, if $v\in\mathbb{C}^\Sigma$ then $v\in[\![\lambda x.Mx]\!]^\Sigma_\rho$ for all M, ρ by axiom (Ax-v): therefore we need $v\in[\![M]\!]^\Sigma_\rho$ for all M, ρ.

($\Leftarrow$) follows from Theorem 27, but for the case $v\in\mathbb{C}^\Sigma$, in which v is harmless being contained in the interpretations of all terms. □

Using Proposition 7(1), Lemma 21 and previous theorems we get:

- $\langle\mathcal{F}^\bigtriangledown,\cdot,[\![\ ]\!]^\bigtriangledown\rangle$ with $\bigtriangledown\in\{\mathcal{B}a,\mathcal{CDV}\}$ is a model of the $\lambda\mathbf{I}$-calculus,
- $\langle\mathcal{F}^\bigtriangledown,\cdot,[\![\ ]\!]^\bigtriangledown\rangle$ with $\bigtriangledown\in\{\mathcal{HL},\mathcal{HR}\}$ is an extensional model of the $\lambda\mathbf{I}$-calculus,
- $\langle\mathcal{F}^{\mathcal{EHR}},\cdot,[\![\ ]\!]^{\mathcal{EHR}}\rangle$ is a model of the "call-by-value" λ-calculus,
- $\langle\mathcal{F}^\bigtriangledown,\cdot,[\![\ ]\!]^\bigtriangledown\rangle$ with $\bigtriangledown\in\{\mathcal{AO},\mathcal{BCD},\mathcal{P}l,\mathcal{E}n\}$ is a model of the whole λ-calculus,
- $\langle\mathcal{F}^\bigtriangledown,\cdot,[\![\ ]\!]^\bigtriangledown\rangle$ with $\bigtriangledown\in\{\mathcal{S}c,\mathcal{P}a,\mathcal{CDZ},\mathcal{DHM}\}$ is an extensional model of the whole λ-calculus.

Let $\Sigma^\diamond$ be the preorder defined in Example 8: Ref. [4] proves that it induces a model of the whole λ-calculus by showing condition (4)(a) of Theorem 37.

6. Conclusion

When stepping into the world of λ-calculus semantics, intersection-type systems turn out to be a useful "vehicle" to move around, since they provide a finitary way to describe and analyse particular classes of models. By simply adding a single constant or condition on a type preorder, a different semantical domain is characterised. One is then naturally induced to expect that intersection types will provide, in the long run, a sort of tailor shop in which particular domains can be tailored for any specific need.

In the present paper, we have provided characterisation results concerning intersection-type systems for the λ-calculus, for a number of its restrictions and for their corresponding extensional versions. Some results characterise those intersection-type systems for which typing invariance holds w.r.t. β- and η-conversion. Filter λ-structures induced by intersection-type preorders have been shown to provide models for the whole λ-calculus and for a number of relevant "restricted" λ-calculi whenever particular conditions on the type preorders are fulfilled. These characterisations have an interest per se in the syntactic theory of intersection types. However, the paper keeps also a general perspective, since expansion/reduction results are parametric over the set of restricted redexes. Therefore, we have a basis for further analysis: whenever operational investigation will point at new sets of restricted redexes, the present paper's results will provide a useful preliminary tool for isolating conversion properties, thus characterising the corresponding filter models.

Acknowledgements

The authors are grateful to Furio Honsell, Henk Barendregt and Wil Dekkers for enlightening discussions on the subject of the present paper. Moreover, they would like to thank the two anonymous referees for their several suggestions about the form of the paper as well as their help in correcting a few flaws in some proofs. The authors thank also the referees of WoLLIC'03 and TYPES'03 for careful reading preliminary versions of this paper and for their useful comments. The second author wishes to express his thanks to Gisella Meli and Alessandro Cavalli.

References

[1] S. Abramsky, Domain theory in logical form, Ann. Pure Appl. Logic 51 (1–2) (1991) 1–77.
[2] S. Abramsky, C.-H.L. Ong, Full abstraction in the lazy lambda calculus, Inform. and Comput. 105 (2) (1993) 159–267.
[3] F. Alessi, F. Barbanera, M. Dezani-Ciancaglini, Intersection types and computational rules, in: R. de Queiroz, E. Pimentel, L. Figueiredo (Eds.), WoLLIC'03, ENTCS, vol. 84, Elsevier, Amsterdam, 2003.

[4] F. Alessi, F. Barbanera, M. Dezani-Ciancaglini, Tailoring filter models, in: S. Berardi, M. Coppo, F. Damiani (Eds.), TYPES'03, Lecture Notes in Computer Science, vol. 3085, Springer, Berlin, 2004, pp. 17–33.
[5] F. Alessi, M. Dezani-Ciancaglini, F. Honsell, Filter models and easy terms, in: A. Restivo, S. Ronchi Della Rocca, L. Roversi (Eds.), ICTCS'01, Lecture Notes in Computer Science, vol. 2202, Springer, Berlin, 2001, pp. 17–37.
[6] F. Alessi, M. Dezani-Ciancaglini, S. Lusin, Intersection types and domain operators, Theoret. Comput. Sci. 316 (1–3) (2004) 25–47.
[7] F. Alessi, S. Lusin, Simple easy terms, in: S. van Bakel (Ed.), ITRS'02, ENTCS, Vol. 70, Elsevier, Amsterdam, 2002.
[8] H.P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, revised ed., North-Holland, Amsterdam, 1984.
[9] H. Barendregt, M. Coppo, M. Dezani-Ciancaglini, A filter lambda model and the completeness of type assignment, J. Symbolic Logic 48 (4) (1983) 931–940.
[10] M. Coppo, M. Dezani-Ciancaglini, B. Venneri, Functional characters of solvable terms, Z. Math. Logik Grundlag. Math. 27 (1) (1981) 45–58.
[11] M. Coppo, M. Dezani-Ciancaglini, M. Zacchi, Type theories, normal forms, and D_∞-lambda-models, Inform. and Comput. 72 (2) (1987) 85–116.
[12] M. Coppo, F. Honsell, M. Dezani-Ciancaglini, G. Longo, Extended type structures and filter lambda models, in: G. Lolli, G. Longo, A. Marcja (Eds.), Logic Colloq. '82, North-Holland, Amsterdam, 1984, pp. 241–262.
[13] H. Curry, R. Feys, Combinatory Logic, Studies in Logic and the Foundations of Mathematics, Vol. I, North-Holland, Amsterdam, 1958.
[14] M. Dezani-Ciancaglini, S. Ghilezan, Two behavioural lambda models, in: H. Geuvers, F. Wiedijk (Eds.), TYPES'02, Lecture Notes in Computer Science, vol. 2646, Springer, Berlin, 2003, pp. 127–147.
[15] M. Dezani-Ciancaglini, S. Ghilezan, S. Likavec, Behavioural inverse limit models, Theoret. Comput. Sci. 316 (1–3) (2004) 49–74.
[16] M. Dezani-Ciancaglini, F. Honsell, F. Alessi, A complete characterization of complete intersection-type preorders, ACM TOCL 4 (1) (2003) 120–147.
[17] M. Dezani-Ciancaglini, F. Honsell, Y. Motohama, Compositional characterization of λ-terms using intersection types, Theoret. Comput. Sci. 340 (3) (2005) 459–495.
[18] L. Egidi, F. Honsell, S. Ronchi Della Rocca, Operational, denotational and logical descriptions: a case study, Fund. Inform. 16 (2) (1992) 149–169.
[19] E. Engeler, Algebras and combinators, Algebra Universalis 13 (3) (1981) 389–392.
[20] G. Gierz, K. Hoffmann, K. Keimel, J. Mislove, D. Scott, A Compendium of Continuous Lattices, Springer, Berlin, 1980.
[21] R. Hindley, G. Longo, Lambda-calculus models and extensionality, Z. Math. Logik Grundlag. Math. 26 (4) (1980) 289–310.
[22] F. Honsell, M. Lenisa, Some results on the full abstraction problem for restricted lambda calculi, in: A.M. Borzyszkowski, S. Sokolowski (Eds.), MFCS'93, Lecture Notes in Computer Science, vol. 711, Springer, Berlin, 1993, pp. 84–104.
[23] F. Honsell, M. Lenisa, Semantical analysis of perpetual strategies in λ-calculus, Theoret. Comput. Sci. 212 (1–2) (1999) 183–209.
[24] F. Honsell, S. Ronchi Della Rocca, An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus, J. Comput. System Sci. 45 (1) (1992) 49–75.
[25] D. Park, The Y-combinator in Scott's λ-calculus models (revised version), Theory of Computation Report 13, Department of Computer Science, University of Warwick, 1976.
[26] G.D. Plotkin, Call-by-name, call-by-value and the λ-calculus, Theoret. Comput. Sci. 1 (2) (1975) 125–159.
[27] G.D. Plotkin, Set-theoretical and other elementary models of the λ-calculus, Theoret. Comput. Sci. 121 (1–2) (1993) 351–409.
[28] D.S. Scott, Continuous lattices, in: F.W. Lawvere (Ed.), Toposes, Algebraic Geometry and Logic, Lecture Notes in Mathematics, vol. 274, Springer, Berlin, 1972, pp. 97–136.
[29] D.S. Scott, Open problem, in: C. Böhm (Ed.), Lambda Calculus and Computer Science Theory, Lecture Notes in Computer Science, vol. 37, Springer, Berlin, 1975, p. 369.
[30] S. van Bakel, Complete restrictions of the intersection type discipline, Theoret. Comput. Sci. 102 (1) (1992) 135–163.
[31] S. Vickers, Topology Via Logic, Cambridge University Press, Cambridge, 1989.