A Complete Characterization of Complete Intersection-Type Preorders

M. DEZANI-CIANCAGLINI
Università di Torino
and
F. HONSELL and F. ALESSI
Università di Udine

We characterize those *type preorders* which yield complete *intersection-type assignment systems* for λ-calculi, with respect to the three canonical set-theoretical semantics for intersection-types: the inference semantics, the simple semantics and the F-semantics. These semantics arise by taking as interpretation of types subsets of applicative structures, as interpretation of the *preorder relation*, $\leq$, set-theoretic inclusion, as interpretation of the *intersection constructor*, $\cap$, set-theoretic intersection, and by taking the interpretation of the *arrow constructor*, $\rightarrow$ *à la* Scott, with respect to either *any possible functionality set*, or the *largest* one, or the *least* one.

These results strengthen and generalize significantly all earlier results in the literature, to our knowledge, in at least three respects. First of all the inference semantics had not been considered before. Secondly, the characterizations are all given just in terms of simple closure conditions on the *preorder relation*, $\leq$, on the types, rather than on the typing judgments themselves. The task of checking the condition is made therefore considerably more tractable. Lastly, we do not restrict attention just to λ-models, but to arbitrary applicative structures which admit an interpretation function. Thus we allow also for the treatment of models of restricted λ-calculi. Nevertheless the characterizations we give can be tailored just to the case of λ-models.

Categories and Subject Descriptors: F.4.1 [**Theory of Computation**]: Mathematical Logic—*Lambda calculus and related systems*; F.3.3 [**Theory of Computation**]: Studies of Program Constructs—*Type structure*; F.3.2 [**Theory of Computation**]: Semantics of Programming Languages—*Denotational semantics*; D.1.1 [**Software**]: Applicative (Functional) Programming

General Terms: Theory, Languages

Additional Key Words and Phrases: Lambda calculus, Intersection Types, Lambda Models, Completness

1. INTRODUCTION

Intersection-types disciplines originated in [Coppo and Dezani-Ciancaglini 1980] to overcome the limitations of Curry's type assignment system and to provide a char-

Author addresses: M. Dezani-Ciancaglini, Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy `dezani@di.unito.it`. F. Honsell and F.Alessi, Dipartimento di Matematica ed Informatica, Università di Udine, Via delle Scienze 208, 33100 Udine, Italy `honsell, alessi@dimi.uniud.it`
Partially supported by MURST Cofin '99 TOSCA Project, FGV '99 and CNR-GNSAGA.
Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 20TBD ACM 1529-3785/TBD/TBD $5.00

acterization of *strongly normalizing terms* of the λ-calculus [Pottinger 1980]. But very early on, the issue of *completeness* became crucial. Intersection-type preorders and filter λ-models have been introduced, in [Barendregt et al. 1983], precisely to achieve the completeness for the type assignment system $\vdash_{\Omega}^{\mathcal{BCD}}$, with respect to Scott's simple semantics. And this result, together with the conservativity of $\vdash_{\Omega}^{\mathcal{BCD}}$, with respect to Curry's simple types, was used in [Barendregt et al. 1983] to prove Scott's conjecture concerning the completeness of the set-theoretic semantics for simple types.

The number of type preorders of interest in the literature has grown considerably over the years (e.g. [Coppo et al. 1984; Coppo et al. 1987; Honsell and Ronchi della Rocca 1992; Egidi et al. 1992; Abramsky and Ong 1993; Plotkin 1993; Honsell and Lenisa 1999], etc.), especially in connection with the study of domain models for λ-calculi in the perspective of Abramsky's "domain theory in logical form" [Abramsky 1991]. Furthermore new semantics have been proposed for intersection-types [Hindley 1983a].

The problem of characterizing syntactically the sound and adequate (complete) type preorders, with respect to the various set-theoretic semantics, appears therefore rather natural. Moreover, we feel that the very existence of completeness results with respect to set-theoretic semantics, such as the one in [Barendregt et al. 1983], is probably one of the most significant features of intersection-types.

In this paper we solve completely the characterization problem as far as the three canonical *set-theoretical* semantics for intersection-types: the inference semantics, the simple semantics [Scott 1975] and the F-semantics [Scott 1980b]. These are the semantics which arise by interpreting types as subsets of applicative structures, and by taking as interpretation of the *preorder relation*, $\leq$, set-theoretic inclusion, as interpretation of the *intersection constructor*, $\cap$, set-theoretic intersection, and by taking the interpretation of the *arrow constructor*, $\rightarrow$ *à la Scott*, as a *logical predicate*, with respect to either *any possible functionality set*, or the *largest* one, or the *least* one.

More precisely, the *simple semantics* of types associates to each arrow type $A \rightarrow B$ the set of elements which applied to an arbitrary element in the interpretation of A return an element in the interpretation of B.

As Scott has pointed out in [Scott 1980b], however, the key to a lambda model is the set of elements in the domain which are *canonical representatives* of functions, i.e. the elements which are meanings of terms starting with an initial abstraction. The F-semantics of types takes therefore as meaning of an arrow type only those elements which behave as expected with respect to application *and* which are also canonical representatives of functions.

The inference semantics is the counterpart of the inference semantics for polymorphic types introduced in [Mitchell 1988], generalized to suitable applicative structures with an interpretation function, called λ-applicative structures. Here the interpretation of arrows is taken with respect to an arbitrary set which includes the canonical representatives of functions.

The results in this paper strengthen and generalize significantly all earlier results in the literature, to our knowledge, in at least four respects. First of all we consider a general definition of type preorders which allow to represent not only all inverse limit models [Coppo et al. 1984], but also all graph models [Berline 2000]. Secondly the inference semantics had not been considered before. Thirdly the characterizations are all given just in terms of simple closure conditions on the *preorder relation* , $\leq$, on the types, rather than on the typing judgments themselves, as had been done earlier [Dezani-Ciancaglini and Margaria 1986]. The task of checking the condition is made therefore considerably more tractable. Lastly we do not restrict attention just to λ-models, but to the more general class of λ-applicative structures. Thus we allow also for the treatment of models of restricted λ-calculi, and most notably models of Plotkin's call-by-value λ_v-calculus [Plotkin 1975], and models of the λ-**I**-**N**-calculus of [Honsell and Lenisa 1999]. Nevertheless the characterizations we give can be tailored just to the case of λ-models.

The paper is organized as follows. In Section 2 we introduce type preorders, various kinds of type assignment systems, and we prove Generation Lemmata for these systems. In Section 3 we introduce the basic semantical structures, with respect to which we shall discuss soundness and completeness of type preorders. In Section 4 we study filter structures and prove the crucial property satisfied by the interpretation function over them. Section 5 is the main section of the paper. After introducing the notions of type interpretation domain and semantic satisfiability for the three semantics under consideration, we give the characterization results. Finally in Section 6 we discuss related results and give some final remarks.

2. INTERSECTION-TYPE PREORDERS AND TYPE ASSIGNMENT SYSTEMS

Intersection-types are syntactical objects which are built inductively by closing a given set C of *type atoms* (constants) under the *function type* constructor $\rightarrow$ and the *intersection* type constructor $\cap$.

Definition 2.1 *Intersection-type Languages.* An *intersection-type language*, over C, denoted by $\mathsf{T} = \mathsf{T}(\mathsf{C})$ is defined by the following abstract syntax:

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

Notice that in the definition above the cardinality of C is the only varying parameter.

Notation 2.2. Upper case Roman letters $A, B, \ldots$, will denote arbitrary types. In writing intersection-types we shall use the following convention: the constructor $\cap$ takes precedence over the constructor $\rightarrow$ and it associates to the right. Moreover $A^n \rightarrow B$ will be short for $\underbrace{A \rightarrow \cdots \rightarrow A}_{n} \rightarrow B$.

Much of the expressive power of intersection-type disciplines comes from the fact that types can be endowed with a *preorder relation*, $\leq$, which induces the structure of a meet semi-lattice with respect to $\cap$. This appears natural especially in the semantical setting of the present paper, where the intended meaning of types are sets of denotations, $\cap$ is interpreted as set-theoretic intersection, and $\leq$ is interpreted as set inclusion.

Moreover sometimes we want the maximal element of all types or the maximal element of all arrow types to be atoms: we call these atoms respectively Ω and ν. The corresponding axioms are (Ω) and (ν).

Axiom (Ω) is particularly meaningful when used in combination with the Ω-type assignment system, which essentially treats Ω as the universal type of all λ-terms (see Definition 2.9).

Axiom (ν) states that ν includes any arrow type. This axiom agrees with the ν-type assignment system, which treats ν as the universal type of all λ-abstractions (see Definition 2.10).

Definition 2.3 *Intersection-type Preorders.* An *(intersection)-type preorder* $(\mathsf{C}, \leq)$ is a binary relation $\leq$ on the intersection-type language $\mathsf{T}(\mathsf{C})$ satisfying the following set of axioms and rules:

$A \leq A$	(refl)	$A \leq A \cap A$	(idem)
$A \cap B \leq A$	(incl$_L$)	$A \cap B \leq B$	(incl$_R$)
if $\Omega \in \mathsf{C}$ $\quad A \leq \Omega$	(Ω)	if $\nu \in \mathsf{C}$ $\quad A \to B \leq \nu$	(ν)
$\dfrac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'}$	(mon)	$\dfrac{A \leq B \quad B \leq C}{A \leq C}$	(trans)

Notation 2.4. We will write $A \sim B$ for $A \leq B$ and $B \leq A$.

Notice that associativity and commutativity of $\cap$ (modulo $\sim$) follow easily from the above axioms and rules.

Notation 2.5. Being $\cap$ commutative and associative, we will write $\bigcap_{i \leq n} A_i$ for $A_1 \cap \ldots \cap A_n$. Similarly we will write $\cap_{i \in I} A_i$ where we convene that I denotes always a finite non-empty set.

All the type preorders considered so far in the literature are defined for languages over finite or countable sets of atoms and they are generated by recursive sets $\bigtriangledown$ of atoms and rules of the shape $A \leq B$. "Generation" is in the sense that $A \leq B$ is true if and only if it can be derived from the axioms and rules of $\bigtriangledown$ together with those in Definition 2.3. Such preorders will be denoted by $(\mathsf{C}^{\bigtriangledown}, \leq_{\bigtriangledown})$. 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 $\bigtriangledown \mapsto \leq_{\bigtriangledown}$ is not injective.

(Ω-η)	$\Omega \leq \Omega \to \Omega$
(Ω-*lazy*)	$A \to B \leq \Omega \to \Omega$
($\to$-$\cap$)	$(A \to B) \cap (A \to C) \leq A \to B \cap C$
(η)	$\dfrac{A' \leq A \quad B \leq B'}{A \to B \leq A' \to B'}$

Fig. 1. Some special purpose axioms and rules concerning $\leq$.

Figure 1 gives examples of some special purpose axioms and rules, and Figure 2 presents the most traditional sets $\bigtriangledown$. The names $\bigtriangledown$ are the initials of the authors who have first considered the λ-model induced by the preorder $(\mathsf{C}^{\bigtriangledown}, \leq_{\bigtriangledown})$. The

$\mathsf{C}^{\mathcal{B}a} = \mathsf{C}_\infty$		$\mathcal{B}a = \{(\rightarrow\text{-}\cap), (\eta)\}$	
$\mathsf{C}^{\mathcal{EHR}} = \{\nu\}$		$\mathcal{EHR} = \mathcal{B}a \cup \{(\nu)\}$	
$\mathsf{C}^{\mathcal{AO}} = \{\Omega\}$		$\mathcal{AO} = \mathcal{B}a \cup \{(\Omega), (\Omega\text{-}lazy)\}$	
$\mathsf{C}^{\mathcal{BCD}} = \{\Omega\} \cup \mathsf{C}_\infty$		$\mathcal{BCD} = \mathcal{B}a \cup \{(\Omega), (\Omega\text{-}\eta)\}$	

Fig. 2. Examples of finitely generated preorders: atoms, axioms and rules.

order is logical, rather than historical: $\mathcal{B}a$ [van Bakel 1992], $\mathcal{EHR}$ [Egidi et al. 1992], $\mathcal{AO}$ [Abramsky and Ong 1993], $\mathcal{BCD}$ [Barendregt et al. 1983].
The symbol C_∞ denotes an infinite set of fresh atoms, i.e. different from Ω, ν.

The meaning of the axioms and rules of Figure 1 can be grasped easily if we consider the intended set-theoretic semantics, whereby types denote subsets of a domain of discourse, and we interpret $A \to B$ as the set of functions which map each element of A into an element of B.

For instance, in combination with Axiom (Ω), Axiom $(\Omega\text{-}\eta)$ expresses the fact that all the objects in our domain of discourse are total functions, i.e. that Ω is equal to $\Omega \to \Omega$ [Barendregt et al. 1983].

However, if we want to capture only those terms which truly represent functions, as it is necessary, for instance, in discussing the lazy λ-calculus [Abramsky and Ong 1993], we cannot assume axiom $(\Omega\text{-}\eta)$ in order to ensure that all functions are total. To this end we can postulate instead the weaker property $(\Omega\text{-}lazy)$. According to the set theoretic semantics, this axiom states, in effect, simply that an element which is a function, (since it maps A into B) maps also the whole universe into itself. Notice that, when the type denoting the whole universe, Ω is in C, the role of ν could be played also by the type $\Omega \to \Omega$, provided that axiom $(\Omega\text{-}lazy)$ is in ∇. For this reason it is of no use to have at the same time in the language both ν and Ω. Hence we impose that the two constants do not occur together in any C.

The set-theoretic meaning of Axiom $(\rightarrow\text{-}\cap)$ is immediate: if a function maps A into B, and 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 [Barendregt et al. 1983].

Rule (η) is also very natural set-theoretically: it asserts the arrow constructor is contravariant in the first argument and covariant in the second one. Namely, 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 [Barendregt et al. 1983].

Now that we have introduced type preorders we have to explain how to capitalize effectively on their expressive power. This is achieved via the crucial notion of *intersection type assignment system*. This is a natural extension of Curry's type assignment type to intersection types. First we need some preliminary definitions and notations.

Notation 2.6. Σ will be short for $(\mathsf{C}, \leq)$ and Σ^∇ for $(\mathsf{C}^\nabla, \leq_\nabla)$.

Definition 2.7. (1) A *basis* over C is a set of statements of the shape $x{:}B$, where $B \in \mathsf{T}(\mathsf{C})$, all whose variables are distinct.

(2) An *intersection-type assignment system* relative to $\Sigma = (\mathsf{C}, \leq)$, denoted by $\lambda\cap^\Sigma$,

is a formal system for deriving judgments of the form $\Gamma \vdash^{\Sigma} M : A$, where the *subject* M is an untyped λ-term, the *predicate* A is in $\mathsf{T}(\mathsf{C})$, and Γ is a basis over C.

(3) We will write $x \in \Gamma$ as short for $\exists A.\ (x{:}A) \in \Gamma$, i.e. x occurs as the subject of an assertion in Γ.

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

We define three kinds of type assignment systems, which correspond to the presence or the absence of the atoms Ω, ν.

The first ones, the Basic Type Assignment Systems, deal with sets of atoms not including Ω, ν.

Definition 2.8 *Basic Type Assignment Systems.*
Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder with $\Omega, \nu \notin \mathsf{C}$. The *basic type assignment system* for Σ, denoted by $\lambda\cap^{\Sigma}_{\mathcal{B}}$, is a formal system for deriving judgments of the shape $\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A$. Its rules are the following:

$$
\begin{array}{ll}
(\mathrm{Ax}) & \dfrac{x{:}A \in \Gamma}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} x{:}A} \\[2ex]
(\to \mathrm{I}) & \dfrac{\Gamma, x{:}A \vdash^{\Sigma}_{\mathcal{B}} M : B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} \lambda x.M : A \to B} \\[2ex]
(\to \mathrm{E}) & \dfrac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \to B \quad \Gamma \vdash^{\Sigma}_{\mathcal{B}} N : A}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} MN : B} \\[2ex]
(\cap \mathrm{I}) & \dfrac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \quad \Gamma \vdash^{\Sigma}_{\mathcal{B}} M : B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \cap B} \\[2ex]
(\leq) & \dfrac{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : A \quad A \leq B}{\Gamma \vdash^{\Sigma}_{\mathcal{B}} M : B}
\end{array}
$$

If $\Omega \in \mathsf{C}$, in line with the intended set-theoretic interpretation of Ω as the universe, we extend the Basic Type Assignment System with a suitable axiom for Ω.

Definition 2.9 *Ω-type Assignment Systems.*
Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder with $\Omega \in \mathsf{C}$. The axioms and rules of the *Ω-type assignment system* (denoted $\lambda\cap^{\Sigma}_{\Omega}$), for deriving judgments of the shape $\Gamma \vdash^{\Sigma}_{\Omega} M : A$, are those of the Basic type Assignment System, together with the further axiom

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

Similarly, if $\nu \in \mathsf{C}$, in line with the intended interpretation of ν as the universe of *abstractions*, we define:

Definition 2.10 *ν-type Assignment Systems.*
Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder with $\nu \in \mathsf{C}$. The axioms and rules of the *ν-type assignment system* (denoted $\lambda\cap^{\Sigma}_{\nu}$), for deriving judgments of the shape $\Gamma \vdash^{\Sigma}_{\nu} M : A$, are those of the Basic Type Assignment System, together with the further axiom

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

Notation 2.11. 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 assume that $\lambda\cap^{\Sigma}$ stands for $\lambda\cap^{\Sigma}_{\Omega}$ whenever $\Omega \in \mathsf{C}$, for $\lambda\cap^{\Sigma}_{\nu}$ whenever $\nu \in \mathsf{C}$, and for $\lambda\cap^{\Sigma}_{\mathcal{B}}$ otherwise. Similarly for $\vdash^{\Sigma}$.

We refer to [Barendregt and al. 200X] for a detailed account on the interest and differences of the three kinds of intersection-type assignment systems introduced above. Here we just recall a few suggestive facts. Thanks to the intersection-type constructor, *general self-application* can be typed in the systems $\lambda\cap^{\Sigma}_{\mathcal{B}}$ while this was not the case in Curry's type assignment. In fact it is easy to prove that $\vdash^{\Sigma}_{\mathcal{B}} \lambda x.xx : (A \to B) \cap A \to B$ whilst $\lambda x.xx$ cannot receive any type in the Curry system. Actually, all strongly normalizing terms are typeable in $\lambda\cap^{\Sigma}_{\mathcal{B}}$ for all Σ. All solvable terms can be typed in $\lambda\cap^{\Sigma}_{\Omega}$ for all Σ with some type not equivalent to Ω. For instance, using axiom (Ax-Ω), the term $(\lambda yx.x)(\Delta\Delta)$, where $\Delta \equiv \lambda x.xx$, can be given type $A \to A$. The systems $\lambda\cap^{\Sigma}_{\nu}$ are appropriate for dealing with Plotkin's call-by-value λ_v-calculus [Plotkin 1975]. Also these systems allow to type non-strongly normalizing terms. For instance, one can prove that the term $(\lambda yx.x)(\lambda z.\Delta\Delta)$ may receive type $A \to A$ for all A. Anyway, notice that, as proved in [Egidi et al. 1992], $(\lambda yx.x)(\Delta\Delta)$ cannot be typed in $\lambda\cap^{\Sigma^{\mathcal{EHR}}}_{\nu}$.

Notice that the structural rules of *(weakening)* and *(strengthening)* are admissible in all $\lambda\cap^{\Sigma}$s:

$$(\textit{weakening})\ \frac{\Gamma \vdash^{\Sigma} M : A}{\Gamma, x : B \vdash^{\Sigma} M : A} \quad (\textit{strengthening})\ \frac{\Gamma \vdash^{\Sigma} M : A}{\Gamma\lceil M \vdash^{\Sigma} M : A},$$

where $\Gamma\lceil M = \{x : B \mid x \in FV(M)\}$.

Notice also that the intersection elimination rules

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

can be proved immediately to be derivable in all $\lambda\cap^{\Sigma}$s.
Moreover, by a straightforward induction on the structure of derivations, one can prove that the rule

$$(\leq \mathrm{L})\ \frac{\Gamma, x{:}B \vdash^{\Sigma} M : A \quad C \leq B}{\Gamma, x{:}C \vdash^{\Sigma} M : A}$$

is admissible in all $\lambda\cap^{\Sigma}$s.

We conclude this section by proving a crucial technical result concerning type preorders, which will be useful in Section 5. It is a form of generation (or inversion) lemma, which provides conditions for "reversing" some of the rules of the type assignment systems $\lambda\cap^{\Sigma}$.

Notation 2.12. When we write "...assume $A{\not\sim}\Omega$..." 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 $\nu{\not\leq}A$ must be checked just for $\vdash^{\Sigma}_{\nu}$.

Moreover we write "the type preorder $\Sigma = (\mathsf{C}, \leq)$ validates $\triangledown$" to mean that $A \leq_{\triangledown} B$ implies $A \leq B$ for all $A, B \in \mathsf{T}(\mathsf{C})$.

THEOREM 2.13 GENERATION LEMMA. *Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder.*

(1) Assume $A{\not\sim}\Omega$. Then the following conditions are equivalent:

(a) $\Gamma \vdash^\Sigma x : A$

(b) $(x{:}B) \in \Gamma$ *and* $B \leq A$ *for some* $B \in \mathsf{T}(\mathsf{C})$.

(2) *Assume* $A \not\sim \Omega$. *Then the following conditions are equivalent:*

(a) $\Gamma \vdash^\Sigma MN : A$

(b) $\Gamma \vdash^\Sigma M : B_i \to C_i$, $\Gamma \vdash^\Sigma N : B_i$, *and* $\bigcap_{i\in I} C_i \leq A$ *for some* I *and* $B_i, C_i \in \mathsf{T}(\mathsf{C})$.

(3) *Assume* $A \not\sim \Omega$ *and let* Σ *validate* $\mathcal{B}a$. *Then the following conditions are equivalent:*

(a) $\Gamma \vdash^\Sigma MN : A$

(b) $\Gamma \vdash^\Sigma M : B \to A$, *and* $\Gamma \vdash^\Sigma N : B$ *for some* $B \in \mathsf{T}(\mathsf{C})$.

(4) *Assume* $\nu \not\leq A$. *Then the following conditions are equivalent:*

(a) $\Gamma \vdash^\Sigma \lambda x.M : A$

(b) $\Gamma, x{:}B_i \vdash^\Sigma M : C_i$, *and* $\bigcap_{i\in I}(B_i \to C_i) \leq A$ *for some* I *and* $B_i, C_i \in \mathsf{T}(\mathsf{C})$.

PROOF. The proof of each $(b) \Rightarrow (a)$ is easy. So we only treat $(a) \Rightarrow (b)$.

(1) Easy by induction on derivations, since only the axioms (Ax), (Ax-Ω), and the rules ($\cap$I), ($\leq$) can be applied. Notice that the condition $A \not\sim \Omega$ implies that $\Gamma \vdash^\Sigma x : A$ cannot be obtained just using axioms (Ax-Ω).

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

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

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

$$\forall i \in I.\ \Gamma \vdash^\Sigma M : B_i \to C_i, \Gamma \vdash^\Sigma N : B_i,\ \ \forall j \in J.\ \Gamma \vdash^\Sigma M : D_j \to E_j, \Gamma \vdash^\Sigma N : D_j,$$

and moreover $\bigcap_{i\in I} C_i \leq A_1$, $\bigcap_{j\in J} E_j \leq A_2$. So we are done since $(\bigcap_{i\in I} C_i) \cap (\bigcap_{j\in J} E_j) \leq A$.

(3) Let I, B_i, C_i be as in (2). Applying rule ($\cap$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 ($\leq$) we have $\Gamma \vdash^\Sigma M : \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$, since

$$\bigcap_{i\in I}(B_i \to C_i) \leq \bigcap_{i\in I}(\bigcap_{i\in I} B_i \to C_i) \leq \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$$

by rule (η) and axiom ($\to$-$\cap$).

We can choose $B = \bigcap_{i\in I} B_i$ and conclude $\Gamma \vdash^\Sigma M : B \to A$ since $\bigcap_{i\in I} C_i \leq A$.

(4) If $A \sim \Omega$ we can choose $B \equiv C \equiv \Omega$. Otherwise $A \not\sim \Omega$ and $\nu \not\leq A$. The proof is by induction on derivations. Notice that $\Gamma \vdash^\Sigma \lambda x.M : A$ cannot be obtained just using axioms (Ax-Ω) or (Ax-ν). The only interesting case is again when $A \equiv A_1 \cap A_2$ and the last rule applied is ($\cap$I):

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

As in the proof of (2) we only consider the case $A_1 \not\sim \Omega$, $\nu \not\leq A_1$, $A_2 \not\sim \Omega$, and $\nu \not\leq A_2$.

By induction there are I, B_i, C_i, J, D_j, E_j such that

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

So we are done since $(\bigcap_{i\in I}(B_i \to C_i)) \cap (\bigcap_{j\in J}(D_j \to E_j)) \leq A$. $\square$

Special cases of this theorem have already appeared in the literature, see [Barendregt et al. 1983; Coppo et al. 1984; Coppo et al. 1987; Honsell and Ronchi della Rocca 1992; Egidi et al. 1992].

3. APPLICATIVE STRUCTURES SUITABLE FOR LAMBDA CALCULUS

In this section we introduce the semantical structures which we will consider in our investigation of soundness and completeness of intersection-type assignment systems.

Definition 3.1 *λ-applicative structure.* A *λ-applicative structure* is a triple $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ such that:

(1) $\langle \mathcal{D}, \cdot \rangle$ is an applicative structure;
(2) $[\![\ ]\!]^{\mathcal{D}} : \Lambda \times \mathsf{Env}_{\mathcal{D}} \to \mathcal{D}$, where $\mathsf{Env}_{\mathcal{D}} = [\mathit{Var} \to \mathcal{D}]$, is a mapping (*interpretation function* for λ-terms) which satisfies the following properties
 (a) $[\![MN]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho} \cdot [\![N]\!]^{\mathcal{D}}_{\rho}$;
 (b) $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda y.M[x := y]]\!]^{\mathcal{D}}_{\rho}$ if $y \notin FV(M)$;
 (c) $(\forall d \in \mathcal{D}.\ [\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} = [\![N]\!]^{\mathcal{D}}_{\rho[x:=d]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\rho}$.

If we compare our definition of λ-applicative structures with that of λ-models (first given in [Hindley and Longo 1980], see also Definition 5.3.1 of [Barendregt 1984] and Definition 11.3 of [Hindley and Seldin 1986]) three conditions are missed:

(1) $[\![x]\!]^{\mathcal{D}}_{\rho} = \rho(x)$;
(2) $[\![M]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho'}$ if $\rho(x) = \rho'(x)$ for all $x \in FV(M)$;
(3) $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} \cdot d = [\![M]\!]^{\mathcal{D}}_{\rho[x:=d]}$.

The absence of conditions (1) and (2) allows us to define the interpretation function on filter structures in such a way it coincides with the set of derivable types (see Proposition 4.5 and Theorem 4.6). We omit conditions (3) for considering also models of restricted λ-calculus (Definition 3.2).

One can easily see that Plotkin's λ-structures, as defined in [Plotkin 1993], are λ-applicative structures. In the next section we will introduce filter structures, which are again λ-applicative structures.

Models of, possibly restricted, λ-calculi, as we commonly know them, can be viewed as special λ-applicative structures.

First we need to give the definition of *restricted* λ-calculus.

Definition 3.2 *Restricted λ-calculus.* Let

$$R \subseteq \{\langle(\lambda x.M)N, M[x := N]\rangle \mid M, N \in \Lambda\}.$$

The *restricted λ-calculus* λ_R is the calculus obtained from the standard λ-calculus, by restricting the β rule to the redexes in R.

Clearly when $R = \beta$, λ_R is the standard λ-calculus. The main examples of *truly* restricted λ-calculi are Plotkin's call-by-value λ_v-calculus [Plotkin 1975] and the λ-$\mathbf{I}$-N-calculus of [Honsell and Lenisa 1999]. Finally we give the crucial definition

Definition 3.3 *(Restricted) λ-model.* A *(restricted) λ-model* for the (restricted) λ-calculus λ_R, is a λ-applicative structure, $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$, which moreover satisfies

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

It is easy to see that all notions of models for, possibly restricted, λ-calculi, based on applicative structures, can be cast in the above setting.

4. FILTER STRUCTURES AND INTERPRETATION OF LAMBDA TERMS

In this section we introduce *filter structures*. These are the basic tool for building λ-applicative structures, in effect λ-models, which realize completeness for type preorders.

Filter structures arise naturally in the context of those generalizations of Stone duality that are used in discussing domain theory in logical form (see [Abramsky 1991], [Coppo et al. 1984], [Vickers 1989]).

This approach provides a conceptually independent semantics to intersection-types, the *lattice semantics*. Types are viewed as *compact elements* of domains. The type Ω denotes the least element, intersections denote joins of compact elements, and arrow types allow to internalize the space of continuous endomorphisms. Following the paradigm of Stone duality, type preorders give rise to *filter structures*, where the interpretation of λ-terms can be given through a finitary logical description.

We start by introducing the notion of *filter* of types. Then we show how to associate to each type preorder its filter structure. This is a λ-applicative structure where the interpretation of a λ-term is given by the filter of the types which can be assigned to it.

Definition 4.1. Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder.

(1) A Σ-filter (or simply filter) is a set $X \subseteq \mathsf{T}(\mathsf{C})$ such that
   (a) if $\Omega \in \mathsf{C}$ then $\Omega \in X$;
   (b) if $A \leq B$ and $A \in X$, then $B \in X$;
   (c) if $A, B \in X$, then $A \cap B \in X$;
(2) $\mathcal{F}^{\Sigma}$ denotes the set of Σ-filters;
(3) if $X \subseteq \mathsf{T}(\mathsf{C})$, $\uparrow X$ denotes the filter generated by X;
(4) a filter is *principal* if it is of the shape $\uparrow \{A\}$, for some type A. We shall denote $\uparrow \{A\}$ simply by $\uparrow A$.

Notice that $\uparrow \emptyset$ is the filter $\uparrow \Omega$, if $\Omega \in \mathsf{C}$, and $\emptyset$ otherwise.

It is not difficult to prove that $\mathcal{F}^{\Sigma}$, ordered by subset inclusion, is a complete lattice, whose bottom element is $\uparrow \emptyset$ and whose top element is $\mathsf{T}(\mathsf{C})$. Moreover if $X, Y \in \mathcal{F}^{\Sigma}$, $X \sqcup Y = \uparrow (X \cup Y)$, $X \sqcap Y = X \cap Y$. The sup of a directed set of filters is the set-theoretic union of filters. The *finite*[1] elements are exactly the filters

[1] An element X is *finite* iff for any directed subset $\mathcal{Y}$, $X \sqsubseteq \bigsqcup \mathcal{Y}$ implies that there exists $Y \in \mathcal{Y}$ such that $X \sqsubseteq Y$.

generated by finite sets of types.

Actually $\mathcal{F}^\Sigma$ is *ω-algebraic*, i.e. its set of finite elements is denumerable, and moreover for each filter X, the set $\mathcal{K}(X)$ of finite elements below X is directed, with sup X itself: $\bigcup \mathcal{K}(X) = X$.

The next step is to define application over sets of filters.

Definition 4.2 *Filter structure.* Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder.

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

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

(2) The maps $F^\Sigma : \mathcal{F}^\Sigma \to [\mathcal{F}^\Sigma \to \mathcal{F}^\Sigma]$ and $G^\Sigma : [\mathcal{F}^\Sigma \to \mathcal{F}^\Sigma] \to \mathcal{F}^\Sigma$ are defined by[2]

$$\begin{array}{rcl} F^\Sigma(X) & = & \lambda\!\!\lambda Y \in \mathcal{F}^\Sigma . X \cdot Y; \\ G^\Sigma(f) & = & \begin{cases} \uparrow \{A \to B \mid B \in f(\uparrow A)\} \cup \uparrow \nu & \text{if } \nu \in \mathsf{C} \\ \uparrow \{A \to B \mid B \in f(\uparrow A)\} & \text{otherwise} \end{cases} \end{array}$$

The triple $\langle \mathcal{F}^\Sigma, F^\Sigma, G^\Sigma \rangle$ is called the *filter structure* over Σ.

Notice that if $\{A \to B \mid B \in f(\uparrow A)\}$ is non-empty and $\nu \in \mathsf{C}$ then it follows $\nu \in \uparrow \{A \to B \mid B \in f(\uparrow A)\}$ by axiom (ν).
The definition of G^Σ, above, appears natural once we recall that axiom (Ax-ν) entails that ν is the universal type of functions.

Arrow types correspond to step functions, and they allow to describe the functional behaviour of filters, in the following sense:

Proposition 4.3. *Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder. For all $X \in \mathcal{F}^\Sigma$ we get*

$$F^\Sigma(X) = \bigsqcup \{\uparrow A \Rightarrow \uparrow B \mid A \to B \in X\},$$

*where $\uparrow A \Rightarrow \uparrow B$ is the step function $\lambda\!\!\lambda X.$ **if** $A \in X$ **then** $\uparrow B$ **else** $\uparrow \emptyset$.*

Proof. We show

$$D \in F^\Sigma(X)(\uparrow C) \;\Leftrightarrow\; D \in (\bigsqcup \{\uparrow A \Rightarrow \uparrow B \mid A \to B \in X\})(\uparrow C).$$

Let $D \not\sim \Omega$, otherwise the thesis is trivial.

$$\begin{array}{rcl} D \in X \cdot \uparrow C & \Leftrightarrow & \exists I, A_i, B_i. C \leq \bigcap_{i \in I} A_i, \bigcap_{i \in I} B_i \leq D \text{ and } \forall i \in I. A_i \to B_i \in X \\ & & \text{by definition of application and of filter} \\ & \Leftrightarrow & \exists I, A_i, B_i. (\uparrow C \Rightarrow \uparrow D) \sqsubseteq \bigsqcup_{i \in I} (\uparrow A_i \Rightarrow \uparrow B_i) \text{ and} \\ & & \forall i \in I. A_i \to B_i \in X \\ & & \text{by definition of step function} \\ & \Leftrightarrow & (\uparrow C \Rightarrow \uparrow D) \sqsubseteq \bigsqcup \{\bigsqcup_{i \in J} (\uparrow A_i \Rightarrow \uparrow B_i) \mid A_i \to B_i \in X, J \text{ finite set}\} \\ & & \text{since } \uparrow C \Rightarrow \uparrow D \text{ is compact and the right-hand side is directed} \\ & \Leftrightarrow & (\uparrow C \Rightarrow \uparrow D) \sqsubseteq \bigsqcup \{(\uparrow A \Rightarrow \uparrow B) \mid A \to B \in X\} \\ & \Leftrightarrow & D \in (\bigsqcup \{(\uparrow A \Rightarrow \uparrow B) \mid A \to B \in X\})(\uparrow C). \quad \square \end{array}$$

The next proposition provides a useful tool for relating arrow types to application.

[2] $\lambda\!\!\lambda$ is an informal λ-notation used to define functions, see [Barendregt 1984] page xiii and [Hindley and Seldin 1986] page 130.

PROPOSITION 4.4. *Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder which validates* $\mathcal{B}a$*, and let* $\Omega \to \Omega \in X$ *if* $\Omega \in \mathsf{C}$*. Then for all* $X \in \mathcal{F}^\Sigma, A, B \in \mathsf{T}(\mathsf{C})$ *we get*

$$B \in X \cdot \uparrow A \textit{ iff } A \to B \in X.$$

PROOF. $(\Rightarrow)$
If $B \sim \Omega$ then $\Omega \to \Omega \leq A \to B$ by rule (η). So $A \to B \in X$ by assumption. Otherwise, by definition of application (Definition 4.2(1)),

$$B \in X \cdot \uparrow A \text{ iff } B \in \uparrow \{D \mid \exists\, C \in \uparrow A.\ C \to D \in X\}.$$

Then there is I and C_i, D_i such that

$$A \leq \bigcap_{i \in I} C_i,\ \bigcap_{i \in I} D_i \leq B \text{ and } C_i \to D_i \in X \text{ for all } i \in I,$$

by definition of filter (Definition 4.1). So we get $A \to B \in X$ by axiom $(\to\text{-}\cap)$ and rule (η).
$(\Leftarrow)$ Trivial. $\square$

Filter structures induce immediately λ-applicative structures.

PROPOSITION 4.5. *Let* $\langle \mathcal{F}^\Sigma, F^\Sigma, G^\Sigma \rangle$ *be a filter structure.*
Let ρ *range over the set of term environments* $\mathsf{Env}_\Sigma = [\mathit{Var} \to \mathcal{F}^\Sigma]$*. Define the interpretation function:* $[\![\ ]\!]^\Sigma : \Lambda \times \mathsf{Env}_\Sigma \to \mathcal{F}^\Sigma$ *as follows:*

—*if there exists* $x \in \mathit{Var}$ *such that* $\rho(x) = \emptyset$*, then* $[\![M]\!]^\Sigma_\rho = \emptyset$*;*
—*otherwise put inductively:*

$$\begin{array}{lcl} [\![x]\!]^\Sigma_\rho & = & \rho(x); \\ [\![MN]\!]^\Sigma_\rho & = & F^\Sigma([\![M]\!]^\Sigma_\rho)([\![N]\!]^\Sigma_\rho); \\ [\![\lambda x.M]\!]^\Sigma_\rho & = & G^\Sigma(\lambda\!\!\lambda X \in \mathcal{F}^\Sigma.[\![M]\!]^\Sigma_{\rho[x:=X]}). \end{array}$$

The triple $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma \rangle$ *is a* λ*-applicative structure.*

The interpretation function of a term coincides with the set of types which are derivable for it. This will be a crucial property in showing completeness using filter structures.

THEOREM 4.6. *Let* $\Sigma = (\mathsf{C}, \leq)$ *be a type preorder and* $\mathsf{T} = \mathsf{T}(\mathsf{C})$*. For any* λ*-term* M *and environment* $\rho : \mathit{Var} \to \mathcal{F}^\Sigma$*,*

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

where $\Gamma \models \rho$ *if and only if for all* $x \in \mathit{Var}$*,* $\rho(x) \neq \emptyset$*, and moreover* $(x : B) \in \Gamma$ *implies* $B \in \rho(x)$*.*

PROOF. The thesis is trivial if $\rho(x) = \emptyset$ for some x. In such a case

$$[\![M]\!]^\Sigma_\rho \ = \ \emptyset \ = \ \{A \in \mathsf{T} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\Sigma M : A\},$$

since for no Γ it holds $\Gamma \models \rho$.
Otherwise we prove the thesis by induction on M. Define

$$\begin{array}{lcl} X_\Omega & = & \text{if } \Omega \in \mathsf{C} \quad \text{then } \uparrow \Omega \quad \text{else } \emptyset; \\ X_\nu & = & \text{if } \nu \in \mathsf{C} \quad \text{then } \uparrow \nu \quad \text{else } \emptyset. \end{array}$$

If $M \equiv x$, then

$$\begin{aligned} [\![x]\!]^\Sigma_\rho &= \rho(x) \\ &= \{A \in \mathsf{T} \mid \exists B \in \rho(x).\ B \leq A\} \\ &= \{A \in \mathsf{T} \mid \exists B \in \rho(x).\ x : B \vdash^\Sigma x : A\} \text{ by Theorem 2.13(1)} \\ &= \{A \in \mathsf{T} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\Sigma x : A\}. \end{aligned}$$

If $M \equiv NL$, then

$$\begin{aligned} [\![NL]\!]^\Sigma_\rho &= [\![N]\!]^\Sigma_\rho \cdot [\![L]\!]^\Sigma_\rho \\ &= \uparrow \{C \mid \exists B \in [\![L]\!]^\Sigma_\rho.\ B \to C \in [\![N]\!]^\Sigma_\rho\} \\ &\qquad \text{by definition of application} \\ &= \{A \in \mathsf{T} \mid \exists I, B_i, C_i.\ B_i \to C_i \in [\![N]\!]^\Sigma_\rho, B_i \in [\![L]\!]^\Sigma_\rho, \\ &\quad \textstyle\bigcap_{i\in I} C_i \leq A\} \cup X_\Omega \\ &= \{A \in \mathsf{T} \mid \exists \Gamma \models \rho, I, B_i, C_i.\ \Gamma \vdash^\Sigma N : B_i \to C_i, \\ &\quad \Gamma \vdash^\Sigma L : B_i, \textstyle\bigcap_{i\in I} C_i \leq A\} \cup \{A \in \mathsf{T} \mid A \sim \Omega\} \\ &\quad \text{by induction, } (\textit{weakening}) \text{ and } (\leq \mathrm{L}) \\ &= \{A \in \mathsf{T} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\Sigma NL : A\} \\ &\qquad \text{by Theorem 2.13(2) and axiom (Ax-}\Omega\text{), rule } (\leq). \end{aligned}$$

If $M \equiv \lambda x.N$, then

$$\begin{aligned} [\![\lambda x.N]\!]^\Sigma_\rho &= G^\Sigma(\lambda\!\!\lambda X \in \mathcal{F}^\Sigma.[\![N]\!]^\Sigma_{\rho[x:=X]}) \\ &= \uparrow \{B \to C \mid C \in [\![N]\!]^\Sigma_{\rho[x:=\uparrow B]}\} \cup X_\nu \\ &\quad \text{by definition of } G^\Sigma \\ &= \{A \in \mathsf{T} \mid \exists \Gamma \models \rho, I, B_i, C_i.\ \Gamma, x : B_i \vdash^\Sigma N : C_i, \\ &\quad \textstyle\bigcap_{i\in I} (B_i \to C_i) \leq A\} \cup \{A \in \mathsf{T} \mid A \sim \nu\} \\ &\quad \text{by induction, } (\textit{weakening}) \text{ and } (\leq \mathrm{L}) \\ &= \{A \in \mathsf{T} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\Sigma \lambda x.N : A\} \\ &\quad \text{by Theorem 2.13(4), axiom (Ax-}\nu\text{), and rule } (\leq). \qquad \square \end{aligned}$$

5. SET-THEORETIC SEMANTICS OF INTERSECTION-TYPES

This is the main section of the paper. Here, we discuss completeness for the three *set-theoretic semantics* of intersection-types mentioned in the introduction. In particular, we characterize those type preorders which induce complete type assignment systems for the *inference*, the *simple* and the *F*-semantics, over λ-applicative structures. As we will see these conditions apply also to the preorders which induce complete systems with respect to the three semantics, over λ-models. We recall that according to these semantics the meaning of types are *subsets* of the universe of discourse, i.e. the applicative structure. The "intersection" type constructor is always interpreted as the *set-theoretic intersection*. While, the "arrow" is interpreted as the set of those points, which belong to a suitable distinguished set, and whose applicative behavior is that of mapping the antecedent of the arrow into the consequent.

As we remarked earlier, the very existence of complete type assignment systems for such semantics over applicative structures is one of the strongest motivations for the whole enterprise of developing a theory of intersection-types.

In discussing completeness, *soundness* is not really an issue, since all type preorders are sound. To achieve *adequacy* and hence *completeness* we have to restrict to two *disjoint* classes of type preorders, namely the *natural* preorders and the *strict* preorders. Filter structures are essential to showing adequacy. In such structures, in fact, the set-theoretic interpretation of a type, as an appropriate subset, is in one-to-one correspondence with the *principal* filter generated by that type.

Definition 5.1.

(1) A type preorder $\Sigma = (\mathsf{C}, \leq)$ is *natural* if $\Omega \in \mathsf{C}$ and it validates $\mathcal{AO}$ as defined in Figure 2.

(2) A type preorder $\Sigma = (\mathsf{C}, \leq)$ is *strict* if $\Omega \notin \mathsf{C}$ and it validates $\mathcal{B}a$ as defined in Figure 2.

Notice that a strict type preorder containing the constant ν validates $\mathcal{EHR}$. All the preorders appearing in Figure 2 are natural, if they contain Ω, and strict otherwise.

Type interpretations can be given on λ-applicative structures once we have fixed a distinguished set of functional objects, Φ. There are various choices for this set. Amongst these there is a *maximal one* and a *minimal* one. The former determines the *simple* semantics, the latter the *F-semantics*.

Definition 5.2 *Type Interpretation Domain.*

(1) A *type interpretation domain* is a quadruple $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}}, \Phi \rangle$ such that:
— $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ is a λ-applicative structure;
— Φ is a subset of $\mathcal{D}$, called the *functionality set*, such that $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} \in \Phi$ for all x, M, ρ.

(2) A type interpretation domain $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}}, \Phi \rangle$ is a *simple interpretation domain* if $\Phi = \mathcal{D}$;

(3) A type interpretation domain $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}}, \Phi \rangle$ is an F-*interpretation domain* if $\Phi = \{d \in \mathcal{D} \mid d = [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho}$ for some $x, M, \rho\}$.

Definition 5.3 *Type Interpretation.* Let $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}}, \Phi \rangle$ be a type interpretation domain. The *type interpretation* $[\![\ ]\!]^{\mathcal{I}}_{\mathcal{V}} : \mathsf{T}(\mathsf{C}) \to \mathsf{P}(\mathcal{D})$ induced by the type environment $\mathcal{V} : \mathsf{C} \to \mathsf{P}(\mathcal{D})$ is defined by:

(1) $[\![\Omega]\!]^{\mathcal{I}}_{\mathcal{V}} = \mathcal{D}$;
(2) $[\![\nu]\!]^{\mathcal{I}}_{\mathcal{V}} = \Phi$;
(3) $[\![\psi]\!]^{\mathcal{I}}_{\mathcal{V}} = \mathcal{V}(\psi)$ if $\psi \in \mathsf{C}$ and $\psi \neq \Omega, \nu$;
(4) $[\![A \to B]\!]^{\mathcal{I}}_{\mathcal{V}} = \{X \in \Phi \mid \forall Y \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}}.\ X \cdot Y \in [\![B]\!]^{\mathcal{I}}_{\mathcal{V}}\}$;
(5) $[\![A \cap B]\!]^{\mathcal{I}}_{\mathcal{V}} = [\![A]\!]^{\mathcal{I}}_{\mathcal{V}} \cap [\![B]\!]^{\mathcal{I}}_{\mathcal{V}}$.

This definition is the counterpart for intersection-types of the *inference semantics* for polymorphic types of [Mitchell 1988], generalized by allowing $\langle \mathcal{D}, \cdot, [\![\ ]\!]^{\mathcal{D}} \rangle$ to be just a λ-applicative structure instead of a λ-model.

Once we fix an applicative structure $\langle \mathcal{D}, \cdot \rangle$, and an interpretation function $[\![\ ]\!]^{\mathcal{D}}$, the above definition depends on the choice of the functionality set Φ and the type environment $\mathcal{V}$. The interpretation of the constants $\{\Omega, \nu\}$ takes into account the corresponding axioms of the type assignment systems.

As we remarked above (see page 9), in the definition of λ-applicative structure, we do not postulate, in general, that $[\![x]\!]^{\mathcal{D}}_{\rho} = \rho(x)$. Nevertheless the class of environments which have this property will be of particular significance (provided that they do not induce trivial interpretations, i.e. interpretations in which all terms are equated). Hence we put

Definition 5.4 *Good environments.* Let $\langle \mathcal{D}, \cdot, [\![\;]\!]^{\mathcal{D}} \rangle$ be a λ-applicative structure. The term environment $\rho : \mathit{Var} \to \mathcal{D}$ is *good* if for all $x \in \mathit{Var}$, we have $[\![x]\!]^{\mathcal{D}}_{\rho} = \rho(x)$ and moreover there exist two terms M, N such that $[\![M]\!]^{\mathcal{D}}_{\rho} \neq [\![N]\!]^{\mathcal{D}}_{\rho}$.

In discussing *sound* type assignment systems we consider only type interpretation domains and type environments which are good (the notion of goodness will depend on the current type preorder and on which kind of semantics we are considering) and which agree with the inclusion relation between types in the following sense:

Definition 5.5. Let $\Sigma = (\mathsf{C}, \leq)$ be a type preorder. A type interpretation domain $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\;]\!]^{\mathcal{D}}, \Phi \rangle$ and a type environment $\mathcal{V} : \mathsf{C} \to \mathsf{P}(\mathcal{D})$

(1) are Σ-*good* if for all $A, B \in \mathsf{T}(\mathsf{C})$:
  (a) for all *good* environments ρ and $d \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}}$, $\rho[x := d]$ is *good*;
  (b) for all *good* environments ρ, terms M and variables x,

$$\forall d \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}}.\ [\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} \in [\![B]\!]^{\mathcal{I}}_{\mathcal{V}} \;\;\Rightarrow\;\; [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} \in [\![A \to B]\!]^{\mathcal{I}}_{\mathcal{V}};$$

(2) are F-Σ-*good* if they are Σ-good and moreover for all *good* environments ρ, variables x, and $A \in \mathsf{T}(\mathsf{C})$

$$[\![x]\!]^{\mathcal{D}}_{\rho} \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}} \cap \Phi \;\;\Rightarrow\;\; [\![\lambda y.xy]\!]^{\mathcal{D}}_{\rho} \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}};$$

(3) *agree* with Σ iff for all $A, B \in \mathsf{T}(\mathsf{C})$:

$$A \leq B \;\;\Rightarrow\;\; [\![A]\!]^{\mathcal{I}}_{\mathcal{V}} \subseteq [\![B]\!]^{\mathcal{I}}_{\mathcal{V}}.$$

Condition (2) of Definition 5.5 is true in particular when $\mathcal{I}$ is an F-interpretation domain such that for all good ρ we get that $\rho(x) \in \Phi$ implies $[\![x]\!]^{\mathcal{D}}_{\rho} = [\![\lambda y.xy]\!]^{\mathcal{D}}_{\rho}$.

Remark that the conditions (1) and (2) of Definition 5.5 are true for all known models of (restricted) λ-calculus ([Hindley and Longo 1980], [Plotkin 1975], [Egidi et al. 1992], [Honsell and Lenisa 1999]).

One can easily see that the following holds:

PROPOSITION 5.6.

(*1*) *All type interpretation domains and all type environments agree with* $\mathcal{AO}$ *and with* $\mathcal{EHR}$.

(*2*) *All simple interpretation domains and all type environments agree with* $\mathcal{BCD}$.

We now introduce formally the three semantics. The definitions follow in a natural way from how we have argued informally so far, but for the restriction (in the definition of $\models^{\Sigma}$) to those type interpretation domains and type environments which are Σ-good (F-Σ-good in the case of F-semantics) and which agree with Σ.

Definition 5.7. Let $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\;]\!]^{\mathcal{D}}, \Phi \rangle$ be a type interpretation domain.

(1) $\mathcal{I}, \rho, \mathcal{V} \models M : A$ iff $[\![M]\!]^{\mathcal{D}}_{\rho} \in [\![A]\!]^{\mathcal{I}}_{\mathcal{V}}$;
(2) $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ iff $\mathcal{I}, \rho, \mathcal{V} \models x : B$ for all $(x{:}B) \in \Gamma$.

Definition 5.8 *Semantic Satisfiability.*

(1) $\Gamma \models_i^\Sigma M : A$ iff $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ implies $\mathcal{I}, \rho, \mathcal{V} \models M : A$ for all Σ-good type interpretation domains $\mathcal{I}$ and type environments $\mathcal{V}$ which moreover agree with Σ, and for all good term environments ρ;
(2) $\Gamma \models_s^\Sigma M : A$ iff $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ implies $\mathcal{I}, \rho, \mathcal{V} \models M : A$ for all Σ-good simple interpretations domains $\mathcal{I}$ and type environments $\mathcal{V}$ which moreover agree with Σ, and for all good term environments ρ;
(3) $\Gamma \models_F^\Sigma M : A$ iff $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ implies $\mathcal{I}, \rho, \mathcal{V} \models M : A$ for all F-Σ-good F-interpretations domains $\mathcal{I}$ and type environments $\mathcal{V}$ which moreover agree with Σ, and for all good term environments ρ.

For example $\not\models_i^\Sigma x : \Omega \to \Omega$, $\models_s^\Sigma x : \Omega \to \Omega$ and $\not\models_F^\Sigma x : \Omega \to \Omega$.

In view of the above definition, we can say that the *inference semantics* is given by $\models_i^\Sigma$, the *simple semantics* by $\models_s^\Sigma$, and the *F-semantics* by $\models_F^\Sigma$. The following proposition is immediate:

PROPOSITION 5.9. *If* $\Gamma \models_i^\Sigma M : A$ *then we have both* $\Gamma \models_s^\Sigma M : A$ *and* $\Gamma \models_F^\Sigma M : A$.

Notation 5.10. We shall denote with $\models^\Sigma$ any of the three $\models_i^\Sigma$, $\models_s^\Sigma$, and $\models_F^\Sigma$.

Derivability in the type system implies semantic satisfiability, as shown in the next theorem. Its proof by induction on derivations is straightforward.

THEOREM 5.11 SOUNDNESS. $\Gamma \vdash^\Sigma M : A$ *implies* $\Gamma \models^\Sigma M : A$.

PROOF. By induction on the derivation of $\Gamma \vdash^\Sigma M : A$ using the definition of type interpretation (Definition 5.3).
Rule $(\to \mathrm{E})$ is sound by definition of λ-applicative structure (Definition 3.1).
The soundness of rule $(\to \mathrm{I})$ follows from the restriction to Σ-good type interpretation domains and type environments (Definition 5.5(1)) and from the definition of functionality set (Definition 5.2(1)).
Rule $(\leq)$ is sound since we consider only type interpretation domains and type environments which agree with Σ (Definition 5.5(3)).
The soundness of the other rules is immediate. $\square$

As regards to adequacy, first we observe that only natural or strict type preorders can be adequate. In particular the model $\mathcal{P}_\omega$ [Scott 1976], Engeler models [Engeler 1981] and those graph models which do not satisfy rule (η) (see [Berline 2000] for a description of these models as webbed models and [Barendregt and al. 200X] for their presentation via type preorders), cannot be adequate.

Remark 5.12. Following a referee's suggestion, we conjecture that all graph models would be complete when using more general notions of type interpretation domain and of type interpretation. More precisely the following extensions are worthy investigating:

—defining the fourth components Φ of type interpretation domains as functions from arrow types to subsets of $\mathcal{D}$ such that $[\![\lambda x.M]\!]_\rho^\mathcal{D} \in \Phi(A \to B)$ for all x, M, ρ, A, B;
—replacing condition (4) of Definition 5.3 with

$$(4') \quad [\![A \to B]\!]_\mathcal{V}^\mathcal{I} = \{X \in \Phi(A \to B) \mid \forall Y \in [\![A]\!]_\mathcal{V}^\mathcal{I}.\ X \cdot Y \in [\![B]\!]_\mathcal{V}^\mathcal{I}\}.$$

PROPOSITION 5.13. *(Adequacy implies naturality or strictness) If* $\Gamma \models^{\Sigma} M : A$ *implies* $\Gamma \vdash^{\Sigma} M : A$ *for all* Γ, M, A*, then* Σ *is a natural or a strict type preorder.*

PROOF. It is easy to verify that the hypothesis forces a type preorder to validate rule (η) and axiom $(\to\text{-}\cap)$, and also axioms (Ω), $(\Omega\text{-}lazy)$ when $\Omega \in \mathsf{C}$, axiom (ν) when $\nu \in \mathsf{C}$. For instance, as regards to axiom $(\to\text{-}\cap)$, consider the basis $\Gamma = \{x{:}(A \to B) \cap (A \to C)\}$. From Definition 5.3 we get $\Gamma \models^{\Sigma} x : A \to B \cap C$. Hence, by hypothesis, we have $\Gamma \vdash^{\Sigma} x : A \to B \cap C$. From Theorem 2.13(1) it must hold $(A \to B) \cap (A \to C) \leq A \to B \cap C$, i.e. axiom $(\to\text{-}\cap)$ must hold. □

Now we shall discuss adequacy for each of the three semantics separately.

First we consider the inference semantics. Our goal is to show that all natural and all strict type preorders are adequate for the inference semantics.
For this proof, we focus on the applicative structure induced by the filter structure $\langle \mathcal{F}^{\Sigma}, F^{\Sigma}, G^{\Sigma} \rangle$, and we put:

Definition 5.14. Let $\Sigma = (\mathsf{C}, \leq)$ be a natural or strict type preorder. Let:

(1) Φ^{Σ} be the functionality set defined by

$$\Phi^{\Sigma} = \begin{cases} \{X \in \mathcal{F}^{\Sigma} \mid \Omega \to \Omega \in X\} & \text{if } \Omega \in \mathsf{C}; \\ \{X \in \mathcal{F}^{\Sigma} \mid \nu \in X\} & \text{if } \nu \in \mathsf{C}; \\ \mathcal{F}^{\Sigma} & \text{otherwise.} \end{cases}$$

(2) $\mathcal{I}^{\Sigma}$ be the type interpretation domain $\langle \mathcal{F}^{\Sigma}, \cdot, [\![\ ]\!]^{\Sigma}, \Phi^{\Sigma} \rangle$.

(3) $\mathcal{V}^{\Sigma} : \mathsf{C} \to \mathsf{P}(\mathcal{F}^{\Sigma})$ be the type environment defined by

$$\mathcal{V}^{\Sigma}(\psi) = \{X \in \mathcal{F}^{\Sigma} \mid \psi \in X\}.$$

(4) $[\![\ ]\!]^{\Sigma} : \mathsf{T}(\mathsf{C}) \to \mathsf{P}(\mathcal{F}^{\Sigma})$ be the mapping $[\![\ ]\!]^{\mathcal{I}^{\Sigma}}_{\mathcal{V}^{\Sigma}}$.

Because of (4) of previous definition, the symbol $[\![\ ]\!]^{\Sigma}$ is overloaded, since it refers both to the term interpretation in a filter structure (see Proposition 4.5) and to the type interpretation induced by a type preorder Σ. Anyway no confusion may arise, since the arguments select the interpretation.

Notice that

— when $\Omega \in \mathsf{C}$ we get $[\![\Omega]\!]^{\Sigma} = \mathcal{F}^{\Sigma} = \{X \in \mathcal{F}^{\Sigma} \mid \Omega \in X\} = \mathcal{V}^{\Sigma}(\Omega)$ and $[\![\Omega \to \Omega]\!]^{\Sigma} = \Phi^{\Sigma} = \{X \in \mathcal{F}^{\Sigma} \mid \Omega \to \Omega \in X\}$;

— when $\nu \in \mathsf{C}$ we get $[\![\nu]\!]^{\Sigma} = \Phi^{\Sigma} = \{X \in \mathcal{F}^{\Sigma} \mid \nu \in X\} = \mathcal{V}^{\Sigma}(\nu)$.

The mapping $[\![\ ]\!]^{\Sigma} : \mathsf{T}(\mathsf{C}) \to \mathsf{P}(\mathcal{F}^{\Sigma})$ has the property of associating to each type A the set of filters which contain A (thus preserving the property through which we define $\mathcal{V}^{\Sigma}$ in the basic case of type constants).

LEMMA 5.15. *Let* Σ *be a natural or strict type preorder then*

$$[\![A]\!]^{\Sigma} = \{X \in \mathcal{F}^{\Sigma} \mid A \in X\}.$$

PROOF. By induction on A. The only interesting case is when A is an arrow type. Remark that if $X \in \mathcal{F}^{\Sigma}$ but $X \notin \Phi^{\Sigma}$ then all types in X are intersections of constant types. In fact if X contains an arrow type, then it contains also $\Omega \to \Omega$

when $\Omega \in \mathsf{C}$ (by rule (Ω-*lazy*)), or ν when $\nu \in \mathsf{C}$ (by rule (ν)), so X belongs to Φ^Σ.
If $A \equiv B \to C$ we have

$$\begin{array}{rcll} [\![B \to C]\!]^\Sigma & = & \{X \in \Phi^\Sigma \mid \forall Y \in [\![B]\!]^\Sigma\ X \cdot Y \in [\![C]\!]^\Sigma\} & \text{by definition} \\ & = & \{X \in \Phi^\Sigma \mid \forall Y. B \in Y \Rightarrow C \in X \cdot Y\} & \text{by induction} \\ & = & \{X \in \Phi^\Sigma \mid C \in X \cdot \uparrow B\} & \text{by monotonicity} \\ & = & \{X \in \Phi^\Sigma \mid B \to C \in X\} & \text{by Proposition 4.4} \\ & & & \text{and the definition of } \Phi^\Sigma \\ & = & \{X \in \mathcal{F}^\Sigma \mid B \to C \in X\} & \text{by above.} \quad \square \end{array}$$

Lemma 5.16. *Let $\Sigma = (\mathsf{C}, \leq)$ be a natural or strict type preorder. Then $\mathcal{I}^\Sigma$, $\mathcal{V}^\Sigma$ are Σ-good and they agree with Σ.*

Proof. $\mathcal{I}^\Sigma$, $\mathcal{V}^\Sigma$ satisfy condition (1a) of Definition 5.5 since $\emptyset \notin [\![A]\!]^\Sigma$ for all $A \in \mathsf{T}(\mathsf{C})$ by Lemma 5.15.
For condition (1b) of Definition 5.5 let $X \in [\![A]\!]^\Sigma$ be such that $[\![M]\!]^\Sigma_{\rho[x:=X]} \in [\![B]\!]^\Sigma$. Then, by Lemma 5.15, $B \in [\![M]\!]^\Sigma_{\rho[x:=X]}$, hence $B \in f(X)$, where we have put $f = \lambda\!\!\lambda X.[\![M]\!]^\Sigma_{\rho[x:=X]}$. Notice that:

$$\begin{array}{rcll} F^\Sigma(G^\Sigma(f)) & = & \bigsqcup\{\uparrow A \Rightarrow \uparrow B \mid A \to B \in G^\Sigma(f)\} & \text{by Proposition 4.3} \\ & \sqsupseteq & \bigsqcup\{\uparrow A \Rightarrow \uparrow B \mid B \in f(\uparrow A)\} & \text{by definition of } G^\Sigma \\ & = & f & \text{by definition of step function.} \end{array}$$

We are done since $F^\Sigma(G^\Sigma(f))(X) = [\![\lambda x.M]\!]^\Sigma_\rho \cdot X$.
Lastly notice that as an immediate consequence of the Lemma 5.15 we get

$$A \leq B \Leftrightarrow \forall X \in \mathcal{F}^\Sigma.[A \in X \Rightarrow B \in X] \Leftrightarrow [\![A]\!]^\Sigma \subseteq [\![B]\!]^\Sigma,$$

and therefore $\mathcal{I}^\Sigma, \mathcal{V}^\Sigma$ agree with Σ. $\square$

Finally we can prove the desired adequacy result.

Theorem 5.17. *(Naturality or strictness imply adequacy) Let $\Sigma = (\mathsf{C}, \leq)$ be a natural or a strict type preorder. Then $\Gamma \models^\Sigma_i M : A$ implies $\Gamma \vdash^\Sigma M : A$.*

Proof. We consider the type interpretation domain $\mathcal{I}^\Sigma$. Let ρ_Γ be the term environment defined by $\rho_\Gamma(x) = \{A \in \mathsf{T}(\mathsf{C}) \mid \Gamma \vdash^\Sigma x : A\}$. It is easy to verify that $\mathcal{I}^\Sigma, \rho_\Gamma, \mathcal{V}^\Sigma \models \Gamma$ and that for all $\Gamma' \models \rho_\Gamma$ we have $\Gamma' \vdash^\Sigma M : A \Rightarrow \Gamma \vdash^\Sigma M : A$. Hence we have:

$$\begin{array}{rcll} \Gamma \models^\Sigma_i M : A & \Rightarrow & [\![M]\!]^\Sigma_{\rho_\Gamma} \in [\![A]\!]^\Sigma & \text{by Lemma 5.16 since } \mathcal{I}^\Sigma, \rho_\Gamma, \mathcal{V}^\Sigma \models \Gamma \\ & \Rightarrow & A \in [\![M]\!]^\Sigma_{\rho_\Gamma} & \text{by Lemma 5.15} \\ & \Rightarrow & \Gamma \vdash^\Sigma M : A & \text{by Theorem 4.6 and the above property.} \quad \square \end{array}$$

Hence, by Proposition 5.13 and Theorem 5.17, all and only the natural or strict type preorders turn out to be complete with respect the inference semantics. There are of course many preorders of interest which do not belong to these classes. For instance the type preorder which induces the filter structure isomorphic to Scott's $\mathcal{P}_\omega$ [Scott 1976] is such a preorder. The reader can see [Barendregt and al. 200X] for more examples.

Notice that the preorders $\Sigma^{\mathcal{AO}}$, $\Sigma^{\mathcal{BCD}}$ induce filter structures which are λ-models [Barendregt et al. 1983], the preorder $\Sigma^{\mathcal{EHR}}$ induces a model for the λ_v-calculus

[Egidi et al. 1992], and the preorder $\Sigma^{\mathcal{B}a}$ induces a model for the λ-$\mathbf{I}$-N-calculus [Honsell and Lenisa 1999]. Hence we have that natural preorders, which induce λ-models, are complete also for the class of λ-models, and strict preorders, which induce models of the other two restricted λ-calculi, are complete for the corresponding classes of models.

Now we characterize those preorders which are complete with respect to the simple semantics.

Theorem 5.18. *(Adequacy for the simple semantics) Let* $\Sigma = (\mathsf{C}, \leq)$ *be a type preorder.* $\Gamma \models_s^\Sigma M : A$ *implies* $\Gamma \vdash^\Sigma M : A$ *iff* Σ *is a natural type preorder which validates axiom* $(\Omega$-$\eta)$ *or a strict type preorder such that* $\nu \notin \mathsf{C}$*.*

Proof. $(\Rightarrow)$ From Proposition 5.13 it follows that Σ is natural or strict.
It is easy to check that if $\Omega \not\sim \Omega \to \Omega$ then simple adequacy fails for $\lambda\cap_\Omega^\Sigma$. We have $\models_s^\Sigma x : \Omega \to \Omega$ since $[\![x]\!]_\rho^\mathcal{D} \cdot d \in \mathcal{D}$ for all $\mathcal{D}, d \in \mathcal{D}$ and $\rho : \mathsf{Env}_\mathcal{D} \to \mathcal{D}$. By Theorem 2.13(1) we can deduce $\vdash_\Omega^\Sigma x : \Omega \to \Omega$ only if $\Omega \sim \Omega \to \Omega$.
Let $\nu \in \mathsf{C}$. We have, for any type interpretation domain $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\ ]\!]^\mathcal{D}, \Phi \rangle$ and $\mathcal{V}$ type environment: $[\![\nu]\!]_\mathcal{V}^\mathcal{I} = \Phi = \mathcal{D}$. Hence $\models_s^\Sigma x : \nu$. But it never holds $\vdash_\nu^\Sigma x : \nu$ by Theorem 2.13(1), hence simple adequacy fails if $\nu \in \mathsf{C}$.
This proves $(\Rightarrow)$.
$(\Leftarrow)$ To prove that $\Gamma \models_s^\Sigma M : A$ implies $\Gamma \vdash^\Sigma M : A$ under the given conditions we use the simple type interpretation domain $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma, \mathcal{F}^\Sigma \rangle$, which is just $\mathcal{I}^\Sigma$ as defined in Definition 5.14, with $\Phi^\Sigma = \mathcal{F}^\Sigma$, being either $\nu \notin \mathsf{C}$ or $\Omega \sim \Omega \to \Omega$. By Lemma 5.15 it follows $[\![A]\!]^\Sigma = \{X \in \mathcal{F}^\Sigma \mid A \in X\}$. So we have that $\Gamma \models_s^\Sigma M : A$ implies $A \in [\![M]\!]_{\rho_\Gamma}^\Sigma$ and we conclude $\Gamma \vdash^\Sigma M : A$ as in the last step of the proof of Theorem 5.17. ☐

Among the type preorders of Figure 2, those adequate for the simple semantics are $\Sigma^{\mathcal{B}a}$ and $\Sigma^{\mathcal{BCD}}$. Other adequate type preorders in the literature are those of [Honsell and Lenisa 1999; Scott 1972; Park 1976; Coppo et al. 1987; Honsell and Ronchi della Rocca 1992]. Non adequate type preorders are all those inducing computationally adequate models for the lazy λ-calculus, e.g. $\Sigma^{\mathcal{AO}}$, or for the call-by-value λ-calculus, e.g. $\Sigma^{\mathcal{EHR}}$. The same argument used for the inference semantics allows to show that the natural preorders mentioned in Theorem 5.18, which induce λ-models, are precisely those which are complete also for the class of λ-models, and the strict preorders, which induce λ_v-models and λ-$\mathbf{I}$-N-models, are complete for the corresponding classes of models. The completeness for the simple semantics of $\lambda\cap_\Omega^\Sigma$ whenever Σ validates $\mathcal{BCD}$ and $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma \rangle$ is a λ-model was proved in [Coppo et al. 1984] using filter models and in [Coppo et al. 1987] using the term model of β-equality.

Finally we turn to the F-semantics. The following definition singles out the type preorders which are adequate for the F-semantics as proved in Theorem 5.28.

Definition 5.19. A type preorder $\Sigma = (\mathsf{C}, \leq)$ is an F-*preorder* iff

(1) either Σ is a natural or a strict type preorder such that $\nu \notin \mathsf{C}$ and for all $\psi \in \mathsf{C}$, $A, B \in \mathsf{T}(\mathsf{C})$, there are $I, A_i, B_i \in \mathsf{T}(\mathsf{C})$ such that

$$\psi \cap (A \to B) \sim \bigcap_{i \in I} (A_i \to B_i);$$

(2) or Σ is a strict type preorder such that $\nu \in \mathsf{C}$ and for all $\psi \in \mathsf{C}$ either $\nu \leq \psi$ or there are $I, A_i, B_i \in \mathsf{T}(\mathsf{C})$ such that

$$\psi \cap \nu \sim \bigcap_{i \in I}(A_i \to B_i).$$

For example $\Sigma^{\mathcal{B}a}$, $\Sigma^{\mathcal{EHR}}$, and $\Sigma^{\mathcal{AO}}$ are F-preorders.
Notice that a natural type preorder Σ which validates axiom $(\Omega\text{-}\eta)$ is an F-preorder iff for all $\psi \in \mathsf{C}$ we get $\psi \sim \bigcap_{i \in I}(A_i \to B_i)$, for some $I, A_i, B_i \in \mathsf{T}(\mathsf{C})$.

Next lemma shows that all types of an F-preorder satisfy the conditions of previous definition.

Lemma 5.20. *Let $\Sigma = (\mathsf{C}, \leq)$ be a F-preorder. Then*

(1) if $\nu \notin \mathsf{C}$, then for all $A, B, C \in \mathsf{T}(\mathsf{C})$,

$$C \cap (A \to B) \sim \bigcap_{i \in I}(A_i \to B_i), \text{ for some } I, A_i, B_i \in \mathsf{T}(\mathsf{C});$$

(2) if $\nu \in \mathsf{C}$, then for all $C \in \mathsf{T}(\mathsf{C})$,

$$\text{either } \nu \leq C \text{ or } C \cap \nu \sim \bigcap_{i \in I}(A_i \to B_i), \text{ for some } I, A_i, B_i \in \mathsf{T}(\mathsf{C}).$$

Proof. We just prove the more difficult case, namely (2). We reason by induction on the structure of C. If $C \in \mathsf{C}$ the thesis is trivial. If $C \equiv D \to E$, then $C \leq \nu$, hence $C \cap \nu \sim D \to E$. If $C \equiv D \cap E$ and $\nu \leq C$ then the thesis is immediate. Otherwise we cannot have both $\nu \leq D$ and $\nu \leq E$. Let us suppose $\nu \not\leq D$. Then, by induction, it follows $D \cap \nu \sim \bigcap_{i \in I}(A_i \to B_i)$, for suitable I and $A_i, B_i \in \mathsf{T}(\mathsf{C})$. Now, if $\nu \leq E$, we get $C \cap \nu \sim D \cap \nu$ and we are done by above. If $\nu \not\leq E$, then, by induction, it follows $E \cap \nu \sim \bigcap_{j \in J}(A'_j \to B'_j)$ for suitable J and $A'_j, B'_j \in \mathsf{T}(\mathsf{C})$. Therefore $C \cap \nu \sim (\bigcap_{i \in I}(A_i \to B_i)) \cap (\bigcap_{j \in J}(A'_j \to B'_j))$. $\square$

Since $\nu \in \mathsf{C}$ implies $A \to B \leq \nu$ for all $A, B \in \mathsf{T}(\mathsf{C})$, we easily get:

Corollary 5.21. *Let $\Sigma = (\mathsf{C}, \leq)$ be a F-preorder. Then for all $A, B, C \in \mathsf{T}(\mathsf{C})$:*

$$C \cap (A \to B) \sim \bigcap_{i \in I}(A_i \to B_i), \text{ for some } I, A_i, B_i \in \mathsf{T}(\mathsf{C}).$$

To discuss F-semantics it is useful to characterize the subset of types which are functional.

Definition 5.22. We define the predicate fun on $\mathsf{T}(\mathsf{C})$ by induction on the structure of types:

(1) $\mathsf{fun}(\psi) = \psi \sim \nu$ or $\psi \sim \bigcap_{i \in I}(A_i \to B_i)$ for some $I, A_i, B_i \in \mathsf{T}(\mathsf{C})$;
(2) $\mathsf{fun}(A \to B) = \mathsf{true}$;
(3) $\mathsf{fun}(A \cap B) = \mathsf{fun}(A)$ or $\mathsf{fun}(B)$.

The following proposition gives an alternative characterization of functional types for F-preorders.

Proposition 5.23. *If $\Sigma = \Sigma(\mathsf{C}, \leq)$ is an F-preorder then $\mathsf{fun}(A)$ iff either $A \sim \nu$ or $A \sim \bigcap_{i \in I}(A_i \to B_i)$ for some $I, A_i, B_i \in \mathsf{T}(\mathsf{C})$.*

Proof. $(\Leftarrow)$ is trivial.
$(\Rightarrow)$ We reason by induction on the structure of A. If $A \in \mathsf{C}$, or $A \equiv B \to C$, or $A \sim \nu$, the thesis follows by definition of $\mathsf{fun}(A)$. Otherwise we have $A \equiv B \cap C$

and either $\mathsf{fun}(B)$ or $\mathsf{fun}(C)$. We assume $\mathsf{fun}(B)$, the case $\mathsf{fun}(C)$ being similar. By induction either $B{\sim}\nu$ or $B{\sim}\bigcap_{i\in I}(A_i \to B_i)$ for some I and $A_i, B_i \in \mathsf{T}(\mathsf{C})$. In the first case
- either $\nu \leq C$ and $A{\sim}C \cap \nu{\sim}\nu$,
- or $A{\sim}C \cap \nu{\sim}\bigcap_{i\in J}(A'_j \to B'_j)$, for some $J, A'_j, B'_j \in \mathsf{T}(\mathsf{C})$ by Lemma 5.20(2).
In the second case it follows $A{\sim}C \cap \bigcap_{i\in I}(A_i \to B_i)$. By choosing an arbitrary $i \in I$ we get

$$C \cap (A_i \to B_i){\sim}\bigcap_{i\in J}(A'_j \to B'_j)$$

for some J and $A'_j, B'_j \in \mathsf{T}(\mathsf{C})$ by Corollary 5.21. So we conclude

$$A{\sim}(\bigcap_{i\in I}(A_i \to B_i)) \cap (\bigcap_{i\in J}(A'_j \to B'_j)). \quad \square$$

To prove adequacy we will again use the filter structure $\langle \mathcal{F}^\Sigma, F^\Sigma, G^\Sigma \rangle$ for defining, as in the previous cases, the type interpretation domain $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma, \Phi^\Sigma_F \rangle$. The definition below differs from Definition 5.14 in that we choose a different functionality set.

Definition 5.24. Let $\Sigma = (\mathsf{C}, \leq)$ be an F-preorder. Let:

(1) Φ^Σ_F be the functionality set defined by

$$\Phi^\Sigma_F = \{X \in \mathcal{F}^\Sigma \mid X = [\![\lambda x.M]\!]^\Sigma_\rho \text{ for some } x, M, \rho\};$$

(2) $\mathcal{I}^\Sigma_F$ be the type interpretation domain $\langle \mathcal{F}^\Sigma, \cdot, [\![\ ]\!]^\Sigma, \Phi^\Sigma_F \rangle$;

(3) $\mathcal{V}^\Sigma_F : \mathsf{C} \to \mathsf{P}(\mathcal{F}^\Sigma)$ be the type environment defined by

$$\mathcal{V}^\Sigma_F(\psi) = \{X \in \mathcal{F}^\Sigma \mid \psi \in X\};$$

(4) $[\![\ ]\!]^\Sigma_F : \mathsf{T}(\mathsf{C}) \to \mathsf{P}(\mathcal{F}^\Sigma)$ be the mapping $[\![\ ]\!]^{\mathcal{I}^\Sigma_F}_{\mathcal{V}^\Sigma_F}$.

When restricting to F-preorders, all filters which contain a functional type belong to the functionality set.

LEMMA 5.25. *Let $\Sigma = (\mathsf{C}, \leq)$ be an F-preorder and $X \in \mathcal{F}^\Sigma$. Then $A \in X$ and $\mathsf{fun}(A)$ imply $X \in \Phi^\Sigma_F$.*

PROOF. We show that under the given conditions

$$X = [\![\lambda y.xy]\!]^\Sigma_{\rho_0} \text{ where } \rho_0(x) = X.$$

Proof of $X \subseteq [\![\lambda y.xy]\!]^\Sigma_{\rho_0}$. Take an arbitrary $B \in X$. Notice that if $\nu \in \mathsf{C}$ then $\mathsf{fun}(A)$ implies $A \leq \nu$ by Proposition 5.23. Moreover $\mathsf{fun}(A)$ implies $\mathsf{fun}(A \cap B)$ by Definition 5.22. Then either $A \cap B{\sim}\nu$ or $A \cap B{\sim}\bigcap_{i\in I}(C_i \to D_i)$ for some $I, C_i, D_i \in \mathsf{T}(\mathsf{C})$ again by Proposition 5.23. In the first case we get $\nu \leq B$ and then $\vdash^\Sigma_\nu \lambda y.xy : B$ by axiom (Ax-ν) and rule $(\leq)$. In the second case we can derive $\{x{:}\bigcap_{i\in I}(C_i \to D_i)\} \vdash^\Sigma \lambda y.xy : \bigcap_{i\in I}(C_i \to D_i)$ using axiom (Ax) and rules $(\leq)$, $(\to \mathrm{E})$, $(\to \mathrm{I})$, and $(\cap \mathrm{I})$. This implies $\{x{:}A \cap B\} \vdash^\Sigma \lambda y.xy : B$ by rules $(\leq)$ and $(\leq \mathrm{L})$. In both cases we conclude $B \in [\![\lambda y.xy]\!]^\Sigma_{\rho_0}$ by Theorem 4.6, since $\emptyset \models \rho_0$ (case $A \cap B{\sim}\nu$) and $\{x{:}A \cap B\} \models \rho_0$ (case $A \cap B{\sim}\bigcap_{i\in I}(C_i \to D_i)$).

Proof of $[\![\lambda y.xy]\!]^\Sigma_{\rho_0} \subseteq X$.
By Theorem 4.6, $B \in [\![\lambda y.xy]\!]^\Sigma_{\rho_0}$ implies $\{x{:}C\} \vdash^\Sigma \lambda y.xy : B$ for some $C \in X$.

If $\Omega \in \mathsf{C}$ and $B{\sim}\Omega$ then $B \in X$ for all X. If $\nu \in \mathsf{C}$ and $\nu \leq B$ then $B \in X$ since $A \leq \nu$ by Proposition 5.23. Otherwise we get $\{x{:}C, y{:}D_i\} \vdash^{\Sigma} xy : E_i$ for some $I, D_i, E_i \in \mathsf{T}(\mathsf{C})$ such that $\bigcap_{i\in I}(D_i \to E_i) \leq B$ by Theorem 2.13(4). This implies $\{x{:}C, y{:}D_i\} \vdash^{\Sigma} x : F_i \to E_i$, $\{x{:}C, y{:}D_i\} \vdash^{\Sigma} y : F_i$ for some $F_i \in \mathsf{T}(\mathsf{C})$ by Theorem 2.13(3). Using Theorem 2.13(1) we have $C \leq F_i \to E_i$ and $D_i \leq F_i$ for all $i \in I$, so we get $C \leq D_i \to E_i$ by rule (η), and we can conclude $C \leq B$, i.e. $B \in X$. $\square$

Lemma 5.26. *Let $\Sigma = (\mathsf{C}, \leq)$ be a F-preorder then*

$$[\![A]\!]^{\Sigma}_{F} \;=\; \{X \in \mathcal{F}^{\Sigma} \mid A \in X\}.$$

Proof. The proof by induction on A is similar to that of Lemma 5.15. All cases are trivial but for ν and arrow types.
If $A \equiv \nu$ let X be any filter in Φ^{Σ}_{F}, that is $X = [\![\lambda x.M]\!]^{\Sigma}_{\rho}$ for some x, M, ρ. Then, by Theorem 4.6, $X = \{B \in \mathsf{T}(\mathsf{C}) \mid \exists \Gamma \models \rho.\ \Gamma \vdash^{\Sigma}_{\nu} \lambda x.M : B\}$. Since $\vdash^{\Sigma}_{\nu} \lambda x.M : \nu$, we have $\nu \in X$. Vice versa, if $\nu \in X$, then by Definition 5.22 $\mathsf{fun}(\nu)$, and so by Lemma 5.25, $X \in \Phi^{\Sigma}_{F}$. We have proved, when $\nu \in \mathsf{C}$, that

$$X \in \Phi^{\Sigma}_{F} \;\Leftrightarrow \nu \in X.$$

Hence $[\![\nu]\!]^{\Sigma}_{F} = \Phi^{\Sigma}_{F} = \{X \in \mathcal{F}^{\Sigma} \mid \nu \in X\}$.
If $A \equiv B \to C$ we have

$$\begin{array}{rcll}
[\![B \to C]\!]^{\Sigma}_{F} & = & \{X \in \Phi^{\Sigma}_{F} \mid C \in X \cdot \uparrow B\} & \text{as in the proof of Lemma 5.15} \\
 & = & \{X \in \Phi^{\Sigma}_{F} \mid B \to C \in X\} & \text{by Proposition 4.4 since,} \\
 & & & \text{when } \Omega \in \mathsf{C}, \text{ Theorem 4.6} \\
 & & & \text{and } X = [\![\lambda x.M]\!]^{\Sigma}_{\rho} \text{ imply } \Omega \to \Omega \in X \\
 & = & \{X \in \mathcal{F}^{\Sigma} \mid B \to C \in X\} & \text{by Lemma 5.25 since } \mathsf{fun}(B \to C). \quad \square
\end{array}$$

Lemma 5.27. *Let $\Sigma = (\mathsf{C}, \leq)$ be an F-preorder. Then $\mathcal{I}^{\Sigma}_{F}$, $\mathcal{V}^{\Sigma}_{F}$ are F-Σ-good and they agree with Σ.*

Proof. We can mimic the proof of Lemma 5.16, using Lemma 5.26 instead of Lemma 5.15, for all points of Definition 5.5 but for (2). So we are left to prove that this last condition holds. The key observation is that by Lemma 5.26 and Theorem 4.6

$$(*)\ \ [\![M]\!]^{\Sigma}_{\rho} \in [\![A]\!]^{\Sigma}_{F} \iff \Gamma \vdash^{\Sigma} M : A \text{ for some basis } \Gamma \text{ such that } \Gamma \models \rho.$$

Let $[\![x]\!]^{\Sigma}_{\rho} \in [\![A]\!]^{\Sigma}_{F} \cap \Phi^{\Sigma}_{F}$. Then $[\![x]\!]^{\Sigma}_{\rho} = [\![\lambda z.M]\!]^{\Sigma}_{\rho'}$ for some z, M, ρ'. By $(*)$ there exists a basis Γ' such that $\Gamma' \models \rho'$ and $\Gamma' \vdash^{\Sigma} \lambda z.M : A$. By Theorem 2.13(4) there are I, A_i, B_i, such that $\Gamma' \vdash^{\Sigma} \lambda z.M : \bigcap_{i\in I}(A_i \to B_i)$ and $\bigcap_{i\in I}(A_i \to B_i) \leq A$. Hence by $(*)$ $[\![\lambda z.M]\!]^{\Sigma}_{\rho'} \in [\![\bigcap_{i\in I}(A_i \to B_i)]\!]^{\Sigma}_{F}$. Since $[\![x]\!]^{\Sigma}_{\rho} = [\![\lambda z.M]\!]^{\Sigma}_{\rho'}$ by $(*)$ there exists a basis $\Gamma \models \rho$, such that $\Gamma \vdash^{\Sigma} x : \bigcap_{i\in I}(A_i \to B_i)$, hence, by applying rules $(\leq)$, $(\to$E$)$, $(\to$I$)$ and $(\cap$I$)$, we get $\Gamma \vdash^{\Sigma} \lambda y.xy : \bigcap_{i\in I}(A_i \to B_i)$. So we have by $(*)$ $[\![\lambda y.xy]\!]^{\Sigma}_{\rho} \in [\![\bigcap_{i\in I}(A_i \to B_i)]\!]^{\Sigma}_{F}$. Since $\mathcal{I}^{\Sigma}_{F}$, $\mathcal{V}^{\Sigma}_{F}$ agree with Σ, we get $[\![\bigcap_{i\in I}(A_i \to B_i)]\!]^{\Sigma}_{F} \subseteq [\![A]\!]^{\Sigma}_{F}$, so we conclude $[\![\lambda y.xy]\!]^{\Sigma}_{\rho} \in [\![A]\!]^{\Sigma}_{F}$. $\square$

Theorem 5.28. *(Adequacy for the F-semantics) $\Gamma \models^{\Sigma}_{F} M : A$ implies $\Gamma \vdash^{\Sigma} M : A$ iff Σ is an F-preorder.*

PROOF. First we check that the given conditions are necessary.
Let $[\![x]\!]^{\mathcal{D}}_{\rho} \in [\![\psi \cap (A \to B)]\!]^{\mathcal{D}}_{\mathcal{V}}$ for some $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\;]\!]^{\mathcal{D}}, \Phi \rangle$, $\mathcal{V}$ which are F-Σ-good, agree with Σ, and some good ρ. Then $[\![x]\!]^{\mathcal{D}}_{\rho} \in \Phi$, since $[\![A \to B]\!]^{\mathcal{D}}_{\mathcal{V}} \subseteq \Phi$. By Definition 5.5(2) it follows $[\![\lambda y.xy]\!]^{\mathcal{D}}_{\rho} \in [\![\psi \cap (A \to B)]\!]^{\mathcal{D}}_{\mathcal{V}}$, which implies

$$x{:}\psi \cap (A \to B) \models^{\Sigma}_{F} \lambda y.xy : \psi \cap (A \to B).$$

By a similar argument we can obtain

$$x{:}\psi \cap \nu \models^{\Sigma}_{F} \lambda y.xy : \psi \cap \nu$$

when $\nu \in \mathsf{C}$. Therefore we have F-adequacy of $\lambda\cap^{\Sigma}$ only if we can prove

$$x : \psi \cap (A \to B) \vdash^{\Sigma} \lambda y.xy : \psi \cap (A \to B)$$

(respectively $x : \psi \cap \nu \vdash^{\Sigma}_{\nu} \lambda y.xy : \psi \cap \nu$ when $\nu \in \mathsf{C}$). Let $\nu \notin \mathsf{C}$.

$$\begin{array}{rl}
x{:}\psi \cap (A \to B) \vdash^{\Sigma} \lambda y.xy : \psi \cap (A \to B) & \Rightarrow\; x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\Sigma} xy : B_i \\
& \quad \text{for some } I, A_i, B_i \in \mathsf{T}(\mathsf{C}) \text{ such that} \\
& \quad \bigcap_{i \in I}(A_i \to B_i) \leq \psi \cap (A \to B)\ (\dagger), \\
& \quad \text{by Theorem 2.13(4)} \\
& \Rightarrow\; x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\Sigma} x : C_i \to B_i \\
& \quad \text{and } x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\Sigma} y : C_i \\
& \quad \text{for some } C_i \in \mathsf{T}(\mathsf{C}) \\
& \quad \text{by Theorem 2.13(3)} \\
& \Rightarrow\; \psi \cap (A \to B) \leq \bigcap_{i \in I}(C_i \to B_i) \\
& \quad \text{and } A_i \leq C_i \text{ by Theorem 2.13(1)} \\
& \Rightarrow\; \psi \cap (A \to B) \leq \bigcap_{i \in I}(A_i \to B_i) \\
& \quad \text{by rule } (\eta).
\end{array}$$

This last judgment along with (†) implies $\psi \cap (A \to B) \sim \bigcap_{i \in I}(A_i \to B_i)$.
Similarly from $x{:}\psi \cap \nu \models^{\Sigma}_{F} \lambda y.xy : \psi \cap \nu$ we can show that either $\nu \leq \psi$ or $\psi \cap \nu \sim \bigcap_{i \in I}(A_i \to B_i)$ when $\nu \in \mathsf{C}$.
For the vice versa, we consider the F-interpretation domain $\mathcal{I}^{\Sigma}_{F}$ and the type environment $\mathcal{V}^{\Sigma}_{F}$ of Definition 5.24. They are F-Σ-good and agree with Σ by Lemma 5.27. By Lemma 5.26 $[\![A]\!]^{\Sigma}_{F} = \{X \in \mathcal{F}^{\Sigma} \mid A \in X\}$. So we have that $\Gamma \models^{\Sigma}_{F} M : A$ implies $\Gamma \vdash^{\Sigma} M : A$ mimicking the proof of Theorem 5.17. □

The preorders $\Sigma^{\mathcal{B}a}$, $\Sigma^{\mathcal{EHR}}$, and $\Sigma^{\mathcal{AO}}$, as well as the type preorders of [Honsell and Lenisa 1999; Scott 1972; Park 1976; Coppo et al. 1987; Honsell and Ronchi della Rocca 1992] are adequate for the F-semantics. Moreover for the last five the simple semantics coincides with the F-semantics. The preorder $\Sigma^{\mathcal{BCD}}$ is an example of a preorder which is not adequate with respect to the F-semantics. The remark concerning λ-models and restricted λ-models made for the inference and the simple semantics, applies also to the F-semantics.

6. RELATED WORK AND FINAL REMARKS

In the literature there are essentially five ways of interpreting Curry's types in a model of the untyped λ-calculus. They differ in the interpretation of the *arrow* type constructor. In what follows we shall mainly follow the terminology of [Hindley 1983a].

The simple and the F-semantics are defined as expected.

Following [Scott 1980a], the *quotient set semantics* takes into account that we want to consider equivalent two functions iff they give equivalent results when applied to equivalent arguments. So types are interpreted as partial equivalence relations of the domain rather than simply as subsets. The arrow constructor is defined as for logical relations: $d \sim_{A \to B} d'$ iff for all c, c' such that $c \sim_A c'$ it holds $d \cdot c \sim_B d' \cdot c'$.

The *F-quotient set semantics* [Scott 1976], modifies the quotient set semantics, in the same way as the F-semantics modifies the simple semantics. Namely it requires that all elements of the domain which are equivalent with respect to an arrow must be canonical representatives of functions.

Finally, Mitchell in [Mitchell 1988] introduces another semantics, which he calls *inference semantics*, in which the interpretation of the arrow must *at least* contain the canonical representatives of functions which behave correctly with respect to the application.

All the above semantics easily extend to *intersection-types* [Dezani-Ciancaglini and Margaria 1986] and to *polymorphic types* [Mitchell 1988].

The crucial question in the semantics of types is the completeness of type assignment systems. Hindley proved in [Hindley 1983a] that Curry's type assignment system is complete for all the mentioned semantics. More specifically [Hindley 1983a] and [Hindley 1983b] show the completeness for the simple semantics and moreover that:

(1) $\Gamma \models_F^\Sigma M : A$ if and only if $\Gamma \models_s^\Sigma M : A$, when A is a Curry type;

(2) the simple semantics is a particular case of the quotient set semantics;

(3) the F-semantics is a particular case of the F-quotient set semantics.

The argument showing points (2) and (3) easily extends to intersection and polymorphic types, so for these type disciplines it is enough to discuss only completeness for the simple and the F-semantics to get completeness results for the quotiented versions. One could define also a quotient version of the inference semantics, but this would be treated similarly.

The completeness with respect to the simple semantics, of various intersection-type assignment systems, over λ-models, has been proved in [Barendregt et al. 1983; Hindley 1982; Coppo et al. 1984; Coppo et al. 1987; van Bakel 1992].

As far as the completeness with respect to the F-semantics of intersection-type assignment systems over λ-models, we can cite [Dezani-Ciancaglini and Margaria 1986], [Yokouchi 1994], [Abramsky and Ong 1993]. In [Dezani-Ciancaglini and Margaria 1986] the type preorders which give λ-models where some filters are never interpretations of λ-abstractions and which are complete for the F-semantics are characterized. More specifically it is shown that a type preorder Σ satisfies the previous conditions if and only if $\Omega \not\sim \Omega \to \Omega$, types are invariant under β-equality of subjects, and moreover the following rule (due to R.Hindley):

$$(\text{Hindley rule}) \quad \frac{\Gamma \vdash^\Sigma M : \psi \cap (\Omega^n \to \Omega) \quad x_i \notin FV(M)\ (1 \leq i \leq n)}{\Gamma \vdash^\Sigma \lambda x_1 \ldots x_n . M x_1 \ldots x_n : \psi}$$

for all $\psi \in \mathsf{C}$, is a derived rule.

Yokouchi in [Yokouchi 1994] shows that if we add two suitable rules (quite similar to Hindley rule) to the intersection-type assignment system of [Coppo et al. 1981] we obtain completeness for the F-semantics.

Abramsky and Ong, in [Abramsky and Ong 1993], prove the completeness of the preorder $\Sigma^{\mathcal{AO}}$, with respect to the F-semantics, over applicative structures with convergence.

We conclude the paper with three final remarks.

If Σ is a natural type preorder which is adequate for the F-semantics, then Hindley's rule is admissible in $\lambda\cap^{\Sigma}_{\Omega}$. This follows from the observation that under the given conditions for all $n \geq 0$ and for all $\psi \in \mathsf{C}$ we can find $I, A_i^{(1)}, \ldots, A_i^{(n)}, B_i$ such that

$$\psi \cap (\Omega^n \to \Omega) \sim \bigcap_{i \in I} (A_i^{(1)} \to \ldots \to A_i^{(n)} \to B_i).$$

We could have used the syntactic approach based on term models introduced in [Hindley 1982] for showing all our adequacy results concerning the simple semantics, but not as far as the inference or the F-semantics. To this end, notice that if $\rho(x) = [M]$ and $M\mathbf{I}$ reduces to an abstraction then a fortiori $M\mathbf{1}$ reduces to an abstraction, where $\mathbf{I} \equiv \lambda x.x$ and $\mathbf{1} \equiv \lambda xy.xy$. Therefore $[\![x\mathbf{1}]\!]_\rho$ is the representative of a function whenever $[\![x\mathbf{I}]\!]_\rho$ is the representative of a function. Now consider the F-preorder $(\{\phi, \Omega\}, \leq_{\mathcal{DM}})$ where $\mathcal{DM} = \mathcal{AO} \cup \{\phi \cap (\Omega \to \Omega) \sim \Omega \to \phi\}$ [Dezani-Ciancaglini and Margaria 1986].

We have

$$x{:}(\phi \to \phi) \to \Omega \to \Omega \vdash^{\Sigma^{\mathcal{DM}}}_{\Omega} x\mathbf{I} : \Omega \to \Omega$$

which implies, by soundness,

$$x{:}(\phi \to \phi) \to \Omega \to \Omega \models^{\Sigma^{\mathcal{DM}}}_{F} x\mathbf{I} : \Omega \to \Omega.$$

By the above we get

$$x{:}(\phi \to \phi) \to \Omega \to \Omega \models^{\Sigma^{\mathcal{DM}}}_{F} x\mathbf{1} : \Omega \to \Omega,$$

but it is easy to check, using the Generation Lemma, that we cannot deduce $x\mathbf{1} : \Omega \to \Omega$ from $x{:}(\phi \to \phi) \to \Omega \to \Omega$. As a matter of fact, the proof of completeness for the F-semantics in [Yokouchi 1994] uses a clever variant of the term model for a λ-calculus with constants. It is not clear to us if this could be adapted to the general case treated here.

It would be nice to investigate independent set-theoretic conditions which imply that a type interpretation and a type environment agree with a type preorder. The canonical example in this sense is the one given by partial applicative structures and the preorder $\Sigma^{\mathcal{EHR}}$.

[Dezani-Ciancaglini et al. 2000] is an extended abstract of the present paper.

ACKNOWLEDGMENT

The authors are grateful to Wil Dekkers, Yoko Motohama, and to the referees of ITRS submission and of the present submission for their useful comments and suggestions. In particular the present general definition of type preorder which

allows to represent all graph models was worked out under the crucial guide of a referee of TOCL submission.

REFERENCES

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

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

Barendregt, H. 1984. *The Lambda Calculus: its Syntax and Semantics*, revised ed. North-Holland, Amsterdam.

Barendregt, H. and al. 200X. *Typed λ-calculus and applications.* North-Holland, Amsterdam. (to appear).

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

Berline, C. 2000. From computation to foundations via functions and application: The λ-calculus and its webbed models. *Theoret. Comput. Sci. 249*, 81–161.

Coppo, M. and Dezani-Ciancaglini, M. 1980. An extension of the basic functionality theory for the λ-calculus. *Notre Dame J. Formal Logic 21,* 4, 685–693.

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

Coppo, M., Dezani-Ciancaglini, M., and Venneri, B. 1981. Functional characters of solvable terms. *Z. Math. Logik Grundlag. Math. 27,* 1, 45–58.

Coppo, M., Dezani-Ciancaglini, M., and Zacchi, M. 1987. Type theories, normal forms, and D_∞-lambda-models. *Inform. and Comput. 72,* 2, 85–116.

Dezani-Ciancaglini, M., Honsell, F., and Alessi, F. 2000. A complete characterization of the complete intersection-type theories. In *Proceedings in Informatics*, J. Rolim, A. Broder, A. Corradini, R. Gorrieri, R. Heckel, J. Hromkovic, U. Vaccaro, and J. Wells, Eds. Vol. 8. ITRS'00 Workshop, Carleton-Scientific, Canada, 287–301.

Dezani-Ciancaglini, M. and Margaria, I. 1986. A characterization of F-complete type assignments. *Theoret. Comput. Sci. 45,* 2, 121–157.

Egidi, L., Honsell, F., and Ronchi della Rocca, S. 1992. Operational, denotational and logical descriptions: a case study. *Fund. Inform. 16,* 2, 149–169.

Engeler, E. 1981. Algebras and combinators. *Algebra Universalis 13,* 3, 389–392.

Hindley, J. and Seldin, J. 1986. *Introduction to Combinators and λ-calculus.* Cambridge University Press, Cambridge.

Hindley, J. R. 1982. The simple semantics for Coppo–Dezani–Sallé types. In *International symposium on programming*, M. Dezani-Ciancaglini and U. Montanari, Eds. Lecture Notes in Computer Science. Springer, Berlin, 212–226.

Hindley, J. R. 1983a. The completeness theorem for typing λ-terms. *Theoret. Comput. Sci. 22*, 1–17.

Hindley, J. R. 1983b. Curry's type-rules are complete with respect to F-semantics too. *Theoret. Comput. Sci. 22*, 127–133.

Hindley, J. R. and Longo, G. 1980. Lambda-calculus models and extensionality. *Z. Math. Logik Grundlag. Math. 26,* 4, 289–310.

Honsell, F. and Lenisa, M. 1999. Semantical analysis of perpetual strategies in λ-calculus. *Theoret. Comput. Sci. 212,* 1-2, 183–209.

Honsell, F. and Ronchi della Rocca, S. 1992. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus. *J. Comput. System Sci. 45,* 1, 49–75.

Mitchell, J. 1988. Polymorphic type inference and containment. *Inform. and Comput. 76*, 211–249.

Park, D. 1976. The **Y**-combinator in Scott's λ-calculus models (revised version). Theory of Computation Report 13, Department of Computer Science, University of Warick.

PLOTKIN, G. D. 1975. Call-by-name, call-by-value and the λ-calculus. *Theoret. Comput. Sci. 1,* 2, 125–159.

PLOTKIN, G. D. 1993. Set-theoretical and other elementary models of the λ-calculus. *Theoret. Comput. Sci. 121,* 1-2, 351–409.

POTTINGER, G. 1980. A type assignment for the strongly normalizable λ-terms. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. Hindley and J. Seldin, Eds. Academic Press, London, 561–577.

SCOTT, D. 1972. Continuous lattices. In *Toposes, Algebraic Geometry and Logic*, F. Lawvere, Ed. Lecture Notes in Mathematics, vol. 274. Springer, Berlin, 97–136.

SCOTT, D. 1975. Open problem. In *Lambda Calculus and Computer Science Theory*, C. Böhm, Ed. Lecture Notes in Computer Science, vol. 37. Springer, Berlin, 369.

SCOTT, D. 1976. Data types as lattices. *SIAM J. Comput. 5,* 3, 522–587.

SCOTT, D. 1980a. Lambda calculus: Some models, some philosophy. In *The Kleene Symposium*, J. Barwise, H. J. Keisler, and K. Kunen, Eds. North-Holland, Amsterdam, 223–265.

SCOTT, D. S. 1980b. Letter to Albert Meyer. CMU.

VAN BAKEL, S. 1992. Complete restrictions of the intersection type discipline. *Theoret. Comput. Sci. 102,* 1, 135–163.

VICKERS, S. 1989. *Topology via logic.* Cambridge University Press, Cambridge.

YOKOUCHI, H. 1994. F-semantics for type assignment systems. *Theoret. Comput. Sci. 129*, 39–77.