A complete characterization of the complete intersection-type theories *

F. Alessi[1], M. Dezani-Ciancaglini[2], F. Honsell[1]

[1] Dipartimento di Matematica ed Informatica
via delle Scienze 208, 33100 Udine, Italy
{alessi,honsell}@dimi.uniud.it
[2] Dipartimento di Informatica
corso Svizzera 185, 10142 Torino, Italy
dezani@di.unito.it

Abstract. We characterize those *intersection-type theories* 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 *intersection constructor*, $\bigcap$, set-theoretic inclusion, 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.

1 Introduction

The intersection-types discipline originated in [6] to overcome the limitations of Curry's type assignment system and to provide a characterization of *strongly normalizing terms* of the λ-calculus. But very early on the issue of *completeness* became crucial. Intersection-type theories and filter λ-models have been introduced, in [5], precisely to achieve the completeness for the type assignment system $\lambda\cap^{\mathcal{BCD}}_{\Omega}$, with respect to Scott's simple semantics. And this result, together with the conservativity of $\lambda\cap^{\mathcal{BCD}}_{\Omega}$, with respect to Curry's simple types, was used in [5] to prove Scott's conjecture concerning the completeness of the set-theoretic semantics for simple types.

The number of type theories of interest in the literature has grown considerably over the years (e.g. [9, 8, 17, 11, 2, 20, 16], etc.), especially in connection with the study of domain models for λ-calculi in the perspective of Abramsky's "domain theory in logical form", [1]. Furthermore new semantics have been proposed for intersection-types [14].

The problem of characterizing syntactically the sound and adequate (complete) intersection-type theories, 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 [5], 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 [22] and the F-semantics [25]. These are the semantics which arise by interpreting types as subsets

* Partially supported by MURST Cofin '99 TOSCA Project and FGV '98.

of applicative structures, and by taking as interpretation of the *intersection constructor*, $\bigcap$, set-theoretic inclusion, 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.

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

As Scott has pointed out [25], 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 the application *and* which are also canonical representatives of functions.

The inference semantics is the counterpart of the inference semantics for polymorphic types introduced in [18], generalized to arbitrary 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 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, as had been done earlier [9, 10]. 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, and most notably models of Plotkin's call-by-value λ_v-calculus, and models of the λ-I-N-calculus of [16]. 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 intersection-type theories, 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 intersection-type theories. 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 theories 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 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:*

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

Notation 2 *Upper case Roman letters i.e.* $A, B, \ldots$*, 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.

Definition 3 (Intersection-type Preorder). *Let* $\mathsf{T} = \mathsf{T}(\mathsf{C})$ *be an intersection-type language. An* intersection-type preorder *over* T *is a binary relation* $\leq$ *on* T *satisfying the following set* $\bigtriangledown^0$ *("nabla-zero") of axioms and rules:*

$$\begin{array}{ll} A \leq A & (refl) \\ A \leq A \cap A & (idem) \\ A \cap B \leq A & (incl_L) \\ A \cap B \leq B & (incl_R) \\ \dfrac{A \leq A' \quad B \leq B'}{A \cap B \leq A' \cap B'} & (mon) \\ \dfrac{A \leq B \quad B \leq C}{A \leq C} & (trans)_{\Box} \end{array}$$

Notation 4 *We will write* $A \sim B$ *for* $A \leq B$ *and* $B \leq A$.

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

Notation 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.*

Possibly effective, syntactical presentations of intersection-type preorders can be given using the notion of *intersection-type theory*. An intersection-type theory includes always the basic set $\bigtriangledown^0$ for $\leq$ and possibly other of special purpose axioms and rules.

Definition 6 (Intersection-type Theories). *Let* $\mathsf{T} = \mathsf{T}(\mathsf{C})$ *be an intersection-type language, and let* $\bigtriangledown$ *be a collection of axioms and rules for deriving judgments of the shape* $A \leq B$, *with* $A, B \in \mathsf{T}$. *The* intersection-type theory $\Sigma(\mathsf{C}, \bigtriangledown)$ *is the set of all judgments* $A \leq B$ *derivable from the axioms and rules in* $\bigtriangledown^0 \cup \bigtriangledown$.$\Box$

Notation 7 *When we consider the intersection-type theory* $\Sigma(\mathsf{C}, \bigtriangledown)$, *we will write*

$$\begin{array}{l} \mathsf{C}^{\bigtriangledown} \text{ for } \mathsf{C}, \\ \mathsf{T}^{\bigtriangledown} \text{ for } \mathsf{T}(\mathsf{C}), \\ \Sigma^{\bigtriangledown} \text{ for } \Sigma(\mathsf{C}, \bigtriangledown). \end{array}$$

Moreover $A \leq_{\bigtriangledown} B$ *will be short for* $(A \leq B) \in \Sigma^{\bigtriangledown}$. *Finally we will write* $A \sim_{\bigtriangledown} B \iff A \leq_{\bigtriangledown} B \leq_{\bigtriangledown} A$.

(Ω)	$A \leq \Omega$
(ν)	$A \to B \leq \nu$
$(\Omega\text{-}\eta)$	$\Omega \leq \Omega \to \Omega$
$(\Omega\text{-}lazy)$	$A \to B \leq \Omega \to \Omega$
$(\to\text{-}\cap)$	$(A \to B) \cap (A \to C) \leq A \to B \cap C$
(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 $\leq$.

In Figure 1 appears a list of special purpose axioms and rules which have been considered in the literature. We give just a few lines of motivation for each.

Axiom (Ω) states that the resulting type preorder has a maximal element. 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 10).

The meaning of the other axioms and rules can be grasped easily if we consider again 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 (Ω-η) expresses the fact that all the objects in our domain of discourse are total functions, i.e. that Ω is equal to $\Omega \to \Omega$ ([5]).

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 [2], we cannot assume axiom (Ω-η) in order to ensure that all functions are total. To this end we can postulate instead the weaker property (Ω-*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.

The set-theoretic meaning of Axiom ($\to$-$\cap$) is immediate: if a function maps A into B, and also A into C, then, actually, it maps the whole A into the intersectionof B and C (i.e. into $B \cap C$), see [5].

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 [5].

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 11). Notice that, when the type denoting the whole universe, Ω is in C^∇, the role of ν could be played also by the type $\Omega \to \Omega$. 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 elements Ω and ν play very special roles in the development of the theory. Therefore we stipulate the following blanket assumptions:

Assumption 1 : if $\Omega \in \mathsf{C}^\nabla$ then $(\Omega) \in \nabla$.
Assumption 2 : if $\nu \in \mathsf{C}^\nabla$ then $(\nu) \in \nabla$.

We introduce in Figure 2 a list of significant intersection-type theories which have been extensively considered in the literature. The order is logical, rather than historical: [3, 11, 2, 5].

We shall write denote such theories as Σ^∇ with various different names ∇, picked for mnemonic reasons. For each such ∇ we specify in Figure 2 the type theory $\Sigma^\nabla = \Sigma(\mathsf{C}, \nabla)$ by giving the set of constants C^∇ and the set ∇ of extra axioms and rules taken from Figure 1. Here C_∞ is an infinite set of fresh atoms, i.e. different from Ω, ν.

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

Fig. 2. Type Theories: atoms, axioms and rules.

Before giving the crucial notion of *intersection-type assignment system*, we need to introduce some preliminary definitions and notations.

Definition 8. *A* $\bigtriangledown$-basis *is a set of statements of the shape* $x{:}B$*, where* $B \in \mathsf{T}^{\bigtriangledown}$*, all whose variables are distinct. An* intersection-type assignment system *relative to* $\Sigma^{\bigtriangledown}$*, denoted by* $\lambda\cap^{\bigtriangledown}$*, is a formal system for deriving judgments of the form* $\Gamma \vdash^{\bigtriangledown} M : A$*, where the* subject M *is an untyped* λ*-term, the* predicate A *is in* $\mathsf{T}^{\bigtriangledown}$*, and* Γ *is a* $\bigtriangledown$*-basis. 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*. We say that a term* M *is* typable *in* $\lambda\cap^{\bigtriangledown}$*, for a given* $\bigtriangledown$*-basis* Γ*, if there is a type* $A \in \mathsf{T}^{\bigtriangledown}$ *such that the judgment* $\Gamma \vdash^{\bigtriangledown} M : A$ *is derivable.* □

Definition 9 (Basic Type Assignment System).
Let $\Sigma^{\bigtriangledown}$ *be a type theory. The* basic type assignment system*, denoted by* $\lambda\cap^{\bigtriangledown}_{\mathcal{B}}$*, is a formal system for deriving judgments of the shape* $\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} M : A$*. Its rules are the following:*

$$(Ax)\quad \frac{x{:}A \in \Gamma}{\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} x{:}A}$$

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

$$(\to \mathrm{E})\quad \frac{\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} M : A \to B \quad \Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} N : A}{\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} MN : B}$$

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

$$(\leq_{\bigtriangledown})\quad \frac{\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} M : A \quad A \leq_{\bigtriangledown} B}{\Gamma \vdash^{\bigtriangledown}_{\mathcal{B}} M : B}$$

□

The Basic Type Assignment System can be extended with other rules according to the set of constants belonging to $\mathsf{C}^{\bigtriangledown}$ and the corresponding axioms and rules in $\Sigma^{\bigtriangledown}$.

If $\Omega \in \mathsf{C}^{\bigtriangledown}$, 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 10 (Ω-type Assignment System).
Let $\Sigma^{\bigtriangledown}$ *be a type theory with* $\Omega \in \mathsf{C}^{\bigtriangledown}$*. The axioms and rules of the* Ω-type assignment system *(denoted* $\lambda\cap^{\bigtriangledown}_{\Omega}$*), are those of the Basic type Assignment System, together with the further axiom*

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

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

Definition 11 (ν-type Assignment System).
Let $\Sigma^{\bigtriangledown}$ *be a type theory with* $\nu \in \mathsf{C}^{\bigtriangledown}$*. The axioms and rules of the* ν-type assignment system *(denoted* $\lambda\cap^{\bigtriangledown}_{\nu}$*), are those of the Basic Type Assignment System, together with the further axiom*

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

For ease of notation, we convene that the symbols Ω and ν are reserved for the distinguished type constants used in the systems $\lambda\cap^{\bigtriangledown}_{\Omega}$ and $\lambda\cap^{\bigtriangledown}_{\nu}$, and hence we forbid $\Omega \in \mathsf{C}^{\bigtriangledown}$ or $\nu \in \mathsf{C}^{\bigtriangledown}$ when we deal with $\lambda\cap^{\bigtriangledown}_{\mathcal{B}}$.

Notation 12 *In the following* $\lambda\cap^{\bigtriangledown}$ *will range over* $\lambda\cap^{\bigtriangledown}_{\mathcal{B}}$*,* $\lambda\cap^{\bigtriangledown}_{\Omega}$ *and* $\lambda\cap^{\bigtriangledown}_{\nu}$*. More precisely we assume that* $\lambda\cap^{\bigtriangledown}$ *stands for* $\lambda\cap^{\bigtriangledown}_{\Omega}$ *whenever* $\Omega \in \mathsf{C}^{\bigtriangledown}$*, for* $\lambda\cap^{\bigtriangledown}_{\nu}$ *whenever* $\nu \in \mathsf{C}^{\bigtriangledown}$*, and for* $\lambda\cap^{\bigtriangledown}_{\mathcal{B}}$ *otherwise. Similarly for* $\vdash^{\bigtriangledown}$*.*

For lack of space, we cannot elaborate on the interest and differences of the three intersection-type assignment systems introduced above. We refer to [4] for a detailed account. Here we just recall a few suggestive facts. Thanks to the intersection-type constructor, *self-application* can be typed in the system $\lambda\cap^{\bigtriangledown}_{\mathcal{B}}$, while this was not the case with Curry's type assignment system. For

instance it is easy to prove that $\vdash^{\nabla}_{\mathcal{B}} \lambda x.xx : (A \to B) \cap A \to B$. Actually, all strongly normalizing terms are typeable in $\lambda\cap^{\nabla}_{\mathcal{B}}$. All solvable terms can be typed in $\lambda\cap^{\nabla}_{\Omega}$ 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 system $\lambda\cap^{\nabla}_{\nu}$ is appropriate for dealing with Plotkin's call-by-value λ_v-calculus. Also this system allows 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 [11], $(\lambda yx.x)(\Delta\Delta)$ cannot be typed in $\lambda\cap^{\nabla}_{\nu}$.

Notice that the structural rule of *weakening* is admissible in all $\lambda\cap^{\nabla}$'s.
Notice also that the intersection elimination rules

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

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

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

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

We conclude this section by proving a crucial technical result concerning intersection-type theories, 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 $\lambda\cap^{\nabla}$,

Notation 13 *When we write "...assume $A\not\sim_{\nabla}\Omega$..." we mean that this condition is always true when we deal with $\vdash^{\nabla}_{\mathcal{B}}$ and $\vdash^{\nabla}_{\nu}$, while it must be checked for $\vdash^{\nabla}_{\Omega}$. Similarly, the condition $\nu\not\leq_{\nabla}A$ must be checked just for $\vdash^{\nabla}_{\nu}$.*

Moreover we write "the type theory Σ^{∇} validates ∇'" to mean that all axioms and rules of ∇' are admissible in Σ^{∇}.

Theorem 1 (Generation Lemma). *Let Σ^{∇} be a type theory.*

1. *Assume $A\not\sim_{\nabla}\Omega$. Then $\Gamma \vdash^{\nabla} x : A$ iff $(x{:}B) \in \Gamma$ and $B \leq_{\nabla} A$ for some $B \in \mathsf{T}^{\nabla}$.*
2. *Assume $A\not\sim_{\nabla}\Omega$. Then $\Gamma \vdash^{\nabla} MN : A$ iff $\Gamma \vdash^{\nabla} M : B_i \to C_i$, $\Gamma \vdash^{\nabla} N : B_i$, and $\bigcap_{i\in I} C_i \leq_{\nabla} A$ for some I non-empty and $B_i, C_i \in \mathsf{T}^{\nabla}$.*
3. *Assume $A\not\sim_{\nabla}\Omega$ and let Σ^{∇} validate $\mathcal{B}a$. Then $\Gamma \vdash^{\nabla} MN : A$ iff $\Gamma \vdash^{\nabla} M : B \to A$, and $\Gamma \vdash^{\nabla} N : B$ for some $B \in \mathsf{T}^{\nabla}$.*
4. *Assume $\nu\not\leq_{\nabla}A$. Then $\Gamma \vdash^{\nabla} \lambda x.M : A$ iff $\Gamma, x{:}B_i \vdash^{\nabla} M : C_i$, and $\bigcap_{i\in I}(B_i \to C_i) \leq_{\nabla} A$ for some non-empty set I and $B_i, C_i \in \mathsf{T}^{\nabla}$.*

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

(1) Easy by induction on derivations, since only the axioms (Ax), (Ax-Ω), and the rules ($\cap$I), ($\leq_{\nabla}$) can be applied. Notice that the condition $A\not\sim_{\nabla}\Omega$ implies that $\Gamma \vdash^{\nabla} x : A$ 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):

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

The condition $A\not\sim_{\nabla}\Omega$ implies that we cannot have $A_1\sim_{\nabla}A_2\sim_{\nabla}\Omega$. We do the proof for $A_1\not\sim_{\nabla}\Omega$ and $A_2\not\sim_{\nabla}\Omega$, the other cases can be treated similarly. By induction there are I, B_i, C_i, J, D_j, E_j such that $\forall i \in I.\ \Gamma \vdash^{\nabla} M : B_i \to C_i$, $\Gamma \vdash^{\nabla} N : B_i$, $\forall j \in J.\ \Gamma \vdash^{\nabla} M : D_j \to E_j$, $\Gamma \vdash^{\nabla} N : D_j$, and moreover $\bigcap_{i\in I} C_i \leq_{\nabla} A_1$, $\bigcap_{j\in J} E_j \leq_{\nabla} A_2$. So we are done since $(\bigcap_{i\in I} C_i) \cap (\bigcap_{j\in J} E_j) \leq_{\nabla} A$.

(3) Let I, B_i, C_i be as in (2). Applying rule ($\cap$I) to $\Gamma \vdash^{\nabla} M : B_i \to C_i$ we can derive $\Gamma \vdash^{\nabla} M : \bigcap_{i\in I}(B_i \to C_i)$, so by ($\leq_{\nabla}$) we have $\Gamma \vdash^{\nabla} M : \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$, since
$\bigcap_{i\in I}(B_i \to C_i) \leq_{\nabla} \bigcap_{i\in I}(\bigcap_{i\in I} B_i \to C_i) \leq_{\nabla} \bigcap_{i\in I} B_i \to \bigcap_{i\in I} C_i$ by rule (η) and axiom ($\to$-$\cap$).

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

(4) If $A \sim_{\nabla} \Omega$ we can choose $B \equiv C \equiv \Omega$. Otherwise $A \not\sim_{\nabla} \Omega$ and $\nu \not\leq_{\nabla} A$. The proof is by induction on derivations. Notice that $\Gamma \vdash^{\nabla} \lambda x.M : A$ 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):

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

As in the proof of (2) we only consider the case $A_1 \not\sim_{\nabla} \Omega$, $\nu \not\leq_{\nabla} A_1$, $A_2 \not\sim_{\nabla} \Omega$, and $\nu \not\leq_{\nabla} A_2$. By induction there are I, B_i, C_i, J, D_j, E_j such that

$$\begin{array}{l} \forall i \in I.\ \Gamma, x{:}B_i \vdash^{\nabla} M : C_i, \quad \forall j \in J.\ \Gamma, x{:}D_j \vdash^{\nabla} M : E_j, \\ \bigcap_{i\in I}(B_i \to C_i) \leq_{\nabla} A_1 \ \ \& \ \ \bigcap_{j\in J}(D_j \to E_j) \leq_{\nabla} A_2. \end{array}$$

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

Special cases of this theorem have already appeared in the literature, see [5, 9, 8, 17, 11].

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 14 (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* $\mathsf{Env}_{\mathcal{D}} = [Var \to \mathcal{D}]$, *is an interpretation function for* λ*-terms.*

λ-applicative structures are those applicative structures, which have just enough structure to interpret the language of λ-calculus.

It is often the case that we want to focus on the class of λ-applicative structures which provide a *compositional* interpretation function. Hence we introduce:

Definition 15 (Compositional λ-applicative structure). *A* compositional λ-applicative structure *is a* λ*-applicative structure, where the interpretation function satisfies the following properties*

1. $[\![MN]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho} \cdot [\![N]\!]^{\mathcal{D}}_{\rho}$;
2. $[\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda y.M[x := y]]\!]^{\mathcal{D}}_{\rho}$ *if* $y \notin FV(M)$;
3. $(\forall d \in \mathcal{D}.\ [\![M]\!]^{\mathcal{D}}_{\rho[x:=d]} = [\![N]\!]^{\mathcal{D}}_{\rho[x:=d]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\rho}$.

One can easily see that Plotkin's λ-structures as defined in [20], are compositional λ-applicative structures. In the next section we will introduce filter structures, which are again compositional λ-applicative structures.

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

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

Definition 16 (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 and the λ-I-N-calculus of [16]. Finally we give the crucial definition

Definition 17 ((restricted) compositional λ-model). *A* (resricted) compositional λ-model *for the (restricted) λ-calculus λ_R, is a compositional 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 intersection-type theories.

Filter structures arise naturally in the context of those generalizations of Stone duality that are used in discussing domain theory in logical form (see [1], [9], [26]).

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 theories 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 theory its filter structure. This is a compositional 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 18. *Let Σ^{∇} be a type theory.*

1. *A ∇-filter (or a filter over T^{∇}) is a set $X \subseteq \mathsf{T}^{\nabla}$ such that*
   (a) if $\Omega \in \mathsf{C}^{\nabla}$ then $\Omega \in X$;
   (b) if $A \leq_{\nabla} B$ and $A \in X$, then $B \in X$;
   (c) if $A, B \in X$, then $A \cap B \in X$;
2. *$\mathcal{F}^{\nabla}$ denotes the set of ∇-filters over T^{∇};*
3. *if $X \subseteq \mathsf{T}^{\nabla}$, $\uparrow X$ 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 $\Omega \in \mathsf{C}^{\nabla}$, and $\emptyset$ otherwise.

It is not difficult to prove that $\mathcal{F}^{\nabla}$, ordered by subset inclusion, is an ω-algebraic complete lattice, whose bottom element is $\uparrow \emptyset$, and whose top element is T^{∇}. Moreover if $X, Y \in \mathcal{F}^{\nabla}$, $X \sqcup Y = \uparrow (X \cup Y)$, $X \sqcap Y = X \cap Y$. The sup of a directed set of filters is the set-theoretic union of filters. The finite elements are exactly the filters generated by finite sets of types.

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

Definition 19 (Filter structure). *Let Σ^{∇} be a type theory.*

1. *Application $\cdot : \mathcal{F}^{\nabla} \times \mathcal{F}^{\nabla} \to \mathcal{F}^{\nabla}$ is defined as*

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

2. *The maps $F^{\nabla} : \mathcal{F}^{\nabla} \to [\mathcal{F}^{\nabla} \to \mathcal{F}^{\nabla}]$ and $G^{\nabla} : [\mathcal{F}^{\nabla} \to \mathcal{F}^{\nabla}] \to \mathcal{F}^{\nabla}$ are defined by:*

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

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

Notice that if $\{A \to B \mid B \in f(\uparrow A)\}$ is non-empty and $\nu \in \mathsf{C}^{\nabla}$[1] then $\nu \in \uparrow \{A \to B \mid B \in f(\uparrow A)\}$. The definition of G^{∇}, above, appears natural once we recall that axiom (Ax-ν) entails that ν is the universal type of functions.

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

Proposition 1. *Let Σ^{∇} be a type theory which validates $\mathcal{B}a$, and let $\Omega \to \Omega \in X$ if $\Omega \in \mathsf{C}^{\nabla}$, then for all $X \in \mathcal{F}^{\nabla}, A, B \in \mathsf{T}^{\nabla}$ we get*

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

Proof. ($\Rightarrow$)

If $B \sim_{\nabla} \Omega$ then $\Omega \to \Omega \leq_{\nabla} A \to B$ by rule (η). So $A \to B \in X$ by assumption. Otherwise, by definition of application (Definition 19(1)), $B \in X \cdot \uparrow A$ iff $B \in \uparrow \{D \mid \exists C \in \uparrow A.\ C \to D \in X\}$.

Then there is a finite set I and types C_i, D_i such that $A \leq_{\nabla} \bigcap_{i \in I} C_i$, $\bigcap_{i \in I} D_i \leq_{\nabla} B$ and $C_i \to D_i \in X$ for all $i \in I$, by definition of ∇-filter (Definition 18). So we get $A \to B \in X$ by axiom ($\to$-$\cap$) and rule (η).

($\Leftarrow$) Trivial. □

Filter structures induce immediately compositional λ-applicative structures.

Proposition 2. *Let $\langle \mathcal{F}^{\nabla}, F^{\nabla}, G^{\nabla} \rangle$ be a filter structure.*

Let ρ range over the set of term environments $\mathsf{Env}^{\nabla}_{\mathcal{F}} = [Var \to \mathcal{F}^{\nabla}]$. Define the interpretation function: $[\![\ ]\!]^{\mathcal{F}^{\nabla}} : \Lambda \times \mathsf{Env}^{\nabla}_{\mathcal{F}} \to \mathcal{F}^{\nabla}$ as follows:

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

$$\begin{aligned} [\![x]\!]^{\mathcal{F}^{\nabla}}_{\rho} &= \rho(x); \\ [\![MN]\!]^{\mathcal{F}^{\nabla}}_{\rho} &= F^{\nabla}([\![M]\!]^{\mathcal{F}^{\nabla}}_{\rho})([\![N]\!]^{\mathcal{F}^{\nabla}}_{\rho}); \\ [\![\lambda x.M]\!]^{\mathcal{F}^{\nabla}}_{\rho} &= G^{\nabla}(\lambda\!\!\lambda d \in \mathcal{D}.[\![M]\!]^{\mathcal{D}}_{\rho[x:=d]}). \end{aligned}$$

The triple $\langle \mathcal{F}^{\nabla}, \cdot, [\![\ ]\!]^{\mathcal{F}^{\nabla}} \rangle$ is a compositional 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 2. *For any λ-term M and environment $\rho : Var \to \mathcal{F}^{\nabla}$,*

$$[\![M]\!]^{\mathcal{F}^{\nabla}}_{\rho} = \{A \in \mathsf{T}^{\nabla} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^{\nabla} M : A\},$$

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

Proof. We prove the thesis by induction on M. Define

$X_{\nu} =$ if $\nu \in \mathsf{C}^{\nabla}$ then $\uparrow \nu$ else $\emptyset$;

$X_{\Omega} =$ if $\Omega \in \mathsf{C}^{\nabla}$ then $\uparrow \Omega$ else $\emptyset$.

If $M \equiv x$, then

$$\begin{aligned} [\![x]\!]^{\mathcal{F}^{\nabla}}_{\rho} &= \rho(x) \\ &= \{A \in \mathsf{T}^{\nabla} \mid \exists B \in \rho(x).\ B \leq_{\nabla} A\} \\ &= \{A \in \mathsf{T}^{\nabla} \mid \exists B \in \rho(x).\ x : B \vdash^{\nabla} x : A\} \text{ by Theorem 1(1)} \\ &= \{A \in \mathsf{T}^{\nabla} \mid \exists \Gamma \models \rho.\ \Gamma \vdash^{\nabla} x : A\}. \end{aligned}$$

If $M \equiv NL$, then

[1] By assumption $\nu \in \mathsf{C}^{\nabla}$ implies that Σ^{∇} validates axiom (ν), see page 4.

$$
\begin{aligned}
[\![NL]\!]^{\mathcal{F}^\nabla}_\rho &= [\![N]\!]^{\mathcal{F}^\nabla}_\rho \cdot [\![L]\!]^{\mathcal{F}^\nabla}_\rho \\
&= \uparrow \{C \mid \exists B \in [\![L]\!]^{\mathcal{F}^\nabla}_\rho . B \to C \in [\![N]\!]^{\mathcal{F}^\nabla}_\rho\} \\
&\quad \text{by definition of application} \\
&= \{A \in \mathsf{T}^\nabla \mid \exists I \neq \emptyset, B_i, C_i.\ B_i \to C_i \in [\![N]\!]^{\mathcal{F}^\nabla}_\rho, B_i \in [\![L]\!]^{\mathcal{F}^\nabla}_\rho, \\
&\quad \textstyle\bigcap_{i\in I} C_i \leq_\nabla A\} \cup X_\Omega \\
&= \{A \in \mathsf{T}^\nabla \mid \exists \Gamma \models \rho, I \neq \emptyset, B_i, C_i.\ \Gamma \vdash^\nabla N : B_i \to C_i, \\
&\quad \Gamma \vdash^\nabla L : B_i, \textstyle\bigcap_{i\in I} C_i \leq_\nabla A\} \cup \{A \in \mathsf{T} \mid A \sim_\nabla \Omega\} \\
&\quad \text{by induction and } (\mathit{weakening}) \\
&= \{A \in \mathsf{T}^\nabla \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\nabla NL : A\} \\
&\quad \text{by Theorem 1(2) and axiom (Ax-}\Omega\text{), rule } (\leq_\nabla).
\end{aligned}
$$

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

$$
\begin{aligned}
[\![\lambda x.N]\!]^{\mathcal{F}^\nabla}_\rho &= G^\nabla(\lambda\!\!\lambda X \in \mathcal{F}^\nabla.[\![N]\!]^{\mathcal{F}^\nabla}_{\rho[x:=X]}) \\
&= \uparrow \{B \to C \mid C \in [\![N]\!]^{\mathcal{F}^\nabla}_{\rho[x:=\uparrow B]}\} \cup X_\nu \\
&\quad \text{by definition of } G^\nabla \\
&= \{A \in \mathsf{T}^\nabla \mid \exists \Gamma \models \rho, I \neq \emptyset, B_i, C_i.\ \Gamma, x : B_i \vdash^\nabla N : C_i, \\
&\quad \textstyle\bigcap_{i\in I} (B_i \to C_i) \leq_\nabla A\} \cup \{A \in \mathsf{T}^\nabla \mid A \sim_\nabla \nu\} \\
&\quad \text{by induction} \\
&= \{A \in \mathsf{T}^\nabla \mid \exists \Gamma \models \rho.\ \Gamma \vdash^\nabla \lambda x.N : A\} \\
&\quad \text{by Theorem 1(4), axiom (Ax-}\nu\text{), and rule } (\leq_\nabla). \ \square
\end{aligned}
$$

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 intersection-type theories 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 intersection-type theories are sound. To achieve *adequacy* and hence *completeness* we have to restrict to two *disjoint* classes of type theories, namely the *natural* theories and the *strict* theories. 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 20.

1. *A type theory Σ^∇ is* strict *if $\Omega \notin \mathsf{C}^\nabla$ and it validates $\mathcal{B}a$ as defined in Figure 2.*
2. *A type theory Σ^∇ is* natural *if $\Omega \in \mathsf{C}^\nabla$ and it validates $\mathcal{AO}$ as defined in Figure 2.* $\square$

Notice that by the blanket assumption, that axiom $(\nu) \in \nabla$ whenever $\nu \in \mathsf{C}^\nabla$ (cf. page 4), a strict type theory containing the constant ν validates $\mathcal{EHR}$. All the theories 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 21 (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 22 (Type Interpretation). *Let* $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\,]\!]^{\mathcal{D}}, \Phi \rangle$ *be a type interpretation domain. The* type interpretation *induced by the type environment* $\mathcal{V} : \mathsf{C}^{\nabla} \to \mathsf{P}(\mathcal{D})$ *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* $\psi \in \mathsf{C}^{\nabla}$ *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 [18], 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.

In discussing *sound* type assignment systems we consider only type interpretation domains and type environments which agree with the inclusion relation between types in the following sense:

Definition 23. *A type interpretation domains* $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\,]\!]^{\mathcal{D}}, \Phi \rangle$ *and a type environment* $\mathcal{V} : \mathsf{C}^{\nabla} \to \mathsf{P}(\mathcal{D})$ agree *with a type theory* Σ^{∇} *iff*

$$A \leq_{\nabla} B \Rightarrow [\![A]\!]^{\mathcal{I}}_{\mathcal{V}} \subseteq [\![B]\!]^{\mathcal{I}}_{\mathcal{V}}.\ \square$$

One can easily see that the following holds:

Proposition 3.

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 are natural given how we have argued informally so far, but for the restriction (in the definition of $\models^{\nabla}$) to those type interpretation domains and type environments which agree with Σ^{∇}.

Definition 24 (Semantic Satisfiability). *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* $(x{:}B) \in \Gamma$;
3. $\Gamma \models^{\nabla}_{i} M : A$ *iff* $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ *implies* $\mathcal{I}, \rho, \mathcal{V} \models M : A$ *for all type interpretation domains* $\mathcal{I}$*, type environments* $\mathcal{V}$ *which agree with* Σ^{∇}*, and for all term environments* ρ;
4. $\Gamma \models^{\nabla}_{s} M : A$ *iff* $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ *implies* $\mathcal{I}, \rho, \mathcal{V} \models M : A$ *for all simple interpretations domains* $\mathcal{I}$*, type environments* $\mathcal{V}$ *which agree with* Σ^{∇}*, and for all term environments* ρ;
5. $\Gamma \models^{\nabla}_{F} M : A$ *iff* $\mathcal{I}, \rho, \mathcal{V} \models \Gamma$ *implies* $\mathcal{I}, \rho, \mathcal{V} \models M : A$ *for all* F*-interpretations domains* $\mathcal{I}$*, type environments* $\mathcal{V}$ *which agree with* Σ^{∇}*, and for all term environments* ρ.

In view of the above definition, we can say that the *inference semantics* is given by $\models^{\nabla}_{i}$, the *simple semantics* by $\models^{\nabla}_{s}$, and the *F-semantics* by $\models^{\nabla}_{F}$. The following proposition is immediate:

Proposition 4. *1. If $\Gamma \models_i^{\nabla} M : A$ then we have both $\Gamma \models_s^{\nabla} M : A$ and $\Gamma \models_F^{\nabla} M : A$.*
2. If for all Γ, M, A, we have that, $\Gamma \models_s^{\nabla} M : A$ iff $\Gamma \models_F^{\nabla} M : A$, then all three semantics coincide.

Notation 25 *We shall denote with $\models^{\nabla}$ any of the three $\models_i^{\nabla}$, $\models_s^{\nabla}$, and $\models_F^{\nabla}$.*

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

Theorem 3 (Soundness). *$\Gamma \vdash^{\nabla} M : A$ implies $\Gamma \models^{\nabla} M : A$.* □

As regards to adequacy, first we observe that only natural or strict type theories can be adequate.

Proposition 5. *(Adequacy implies naturality or strictness) If $\Gamma \models^{\nabla} M : A$ implies $\Gamma \vdash^{\nabla} M : A$ for all Γ, M, A, then Σ^{∇} is a natural or a strict type theory.*

Proof. By Proposition 3(1) all type interpretations validate rule (η) and axiom $(\to\text{-}\cap)$, and also axioms (Ω), $(\Omega\text{-}lazy)$ when $\Omega \in \mathsf{C}^{\nabla}$, axiom (ν) when $\nu \in \mathsf{C}^{\nabla}$. □

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

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

Definition 26. *Let Σ^{∇} be a natural or strict type theory. Let:*

1. Φ^{∇} be the functionality set defined by

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

2. $\mathcal{I}^{\nabla}$ be the type interpretation domain $\langle \mathcal{F}^{\nabla}, \cdot, [\![\,]\!]^{\nabla}, \Phi^{\nabla} \rangle$.
3. $\mathcal{V}^{\nabla} : \mathsf{C}^{\nabla} \to \mathsf{P}(\mathcal{F}^{\nabla})$ be the type environment defined by

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

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

Notice that

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

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

Lemma 1. *Let Σ^{∇} be a natural or strict type theory then*

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

Proof. By induction on A. The only interesting case is when A is an arrow type. Remark that if $X \in \mathcal{F}^\nabla$ but $X \notin \Phi^\nabla$ 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 $\Omega \in \mathsf{C}^\nabla$ (by rule (Ω-*lazy*)), or ν when $\nu \in \mathsf{C}^\nabla$ (by rule (ν)), so X belongs to Φ^∇.
If $A \equiv B \to C$ we have

$$\begin{array}{rll}
[\![B \to C]\!]^\nabla & = \{X \in \Phi^\nabla \mid \forall Y \in [\![B]\!]^\nabla\ X \cdot Y \in [\![C]\!]^\nabla\} & \text{by definition} \\
& = \{X \in \Phi^\nabla \mid \forall Y.B \in Y \Rightarrow C \in X \cdot Y\} & \text{by induction} \\
& = \{X \in \Phi^\nabla \mid C \in X \cdot \uparrow B\} & \text{by monotonicity} \\
& = \{X \in \Phi^\nabla \mid B \to C \in X\} & \text{by Proposition 1} \\
& & \text{and the definition of } \Phi^\nabla \\
& = \{X \in \mathcal{F}^\nabla \mid B \to C \in X\} & \text{by above.}\Box
\end{array}$$

As an immediate consequence of the above Lemma we get

$$A \leq_\nabla B \Leftrightarrow \forall X \in \mathcal{F}^\nabla.[A \in X \Rightarrow B \in X] \Leftrightarrow [\![A]\!]^\nabla \subseteq [\![B]\!]^\nabla$$

and therefore $\mathcal{I}^\nabla, \mathcal{V}^\nabla$ agree with Σ^∇.

Finally we can prove the desired adequacy result.

Theorem 4. *(Naturality or strictness imply adequacy) Let Σ^∇ be a natural or a strict type theory, then $\Gamma \models_i^\nabla M : A$ implies $\Gamma \vdash^\nabla M : A$.*

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

$$\begin{array}{rll}
\Gamma \models_i^\nabla M : A & \Rightarrow [\![M]\!]^\nabla_{\rho_\Gamma} \in [\![A]\!]^\nabla & \text{since } \mathcal{I}^\nabla, \rho_\Gamma, \mathcal{V}^\nabla \models \Gamma \\
& \Rightarrow A \in [\![M]\!]^\nabla_{\rho_\Gamma} & \text{by Lemma 1} \\
& \Rightarrow \Gamma \vdash^\nabla M : A & \text{by Theorem 2 and the above property. } \Box
\end{array}$$

Hence the only type theories which turn out to be complete with respect to the inference semantics are the natural and the strict type theories. There are of course many theories of interest which do not belong to these classes. For instance the intersection-type theory which induces the filter structure isomorphic to Scott's $\mathcal{P}(\omega)$ is such a theory. The reader can see [4] for more examples.

Notice that the theory $\mathcal{BCD}$ induces a filter structure which is a λ-model [5], the theory $\mathcal{EHR}$ induces a model for the λ_v-calculus [11], and the theory $\mathcal{B}a$ induces a model for the λ-I-N-calculus [16]. Hence we have that natural theories, which induce λ-models, are complete also for the class of λ-models, and strict theories, which induce the other two restricted λ-calculi, are complete for the corresponding classes of models.

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

Theorem 5. *(Adequacy for the simple semantics) $\Gamma \models_s^\nabla M : A$ implies $\Gamma \vdash^\nabla M : A$ iff Σ^∇ is a strict type theory or a natural type theory which validates axiom (Ω-η).*

Proof. It is easy to check that if $\Omega \not\sim_\nabla \Omega \to \Omega$ then simple adequacy fails for $\lambda\cap^\nabla_\Omega$. We have $\models_s^\nabla x : \Omega \to \Omega$ since $[\![x]\!]^{\mathcal{D}}_\rho \cdot d \in \mathcal{D}$ for all $\mathcal{D}, d \in \mathcal{D}$ and $\rho : \mathsf{Env}_\mathcal{D} \to \mathcal{D}$. But we can deduce $\vdash^\nabla_\Omega x : \Omega \to \Omega$ only if $\Omega \sim_\nabla \Omega \to \Omega$.

To prove that $\Gamma \models_s^\nabla M : A$ implies $\Gamma \vdash^\nabla M : A$ under the given conditions we define the type interpretation domain $\mathcal{I}_s = \langle \mathcal{F}^\nabla, \cdot, [\![\,]\!]^\nabla, \mathcal{F}^\nabla \rangle$ and we use the mapping $[\![\,]\!]^{\mathcal{I}_s}_{\mathcal{V}^\nabla} : \mathsf{T}^\nabla \to \mathsf{P}(\mathcal{F}^\nabla)$ where $\mathcal{V}^\nabla$ is given in Definition 26(3). The proof that $[\![A]\!]^{\mathcal{I}_s}_{\mathcal{V}^\nabla} = \{X \in \mathcal{F}^\nabla \mid A \in X\}$ is similar and simpler than that of Lemma 1. Finally, we conclude mimicking the proof of Theorem 4. $\Box$

Among the type theories of Figure 2, those adequate for the simple semantics are $\mathcal{B}a$ and $\mathcal{BCD}$. Other adequate type theories in the literature are those of [16, 11, 21, 19, 8, 17]. Non adequate type theories are all those inducing computationally adequate models for the lazy λ-calculus, e.g. $\mathcal{AO}$. The same argument used for the inference semantics allows to show that the natural theories mentioned in Theorem 5, which induce λ-models, are precisely those which are complete also for the class of λ-models, and the strict theories, which induce λ_v-models and λ-I-N-models, are complete for the corresponding classes of models. The completeness for the simple semantics of $\lambda\cap^{\nabla}_{\Omega}$ whenever Σ^{∇} validates $\mathcal{BCD}$ and $\langle \mathcal{F}^{\nabla}, \cdot, [\![\,]\!]_{\nabla}\rangle$ is a λ-model was proved in [9] using filter models and in [8] using the term model of β-equality.

Finally we turn to the F-semantics. A first observation is that an F-interpretation $\mathcal{I} = \langle \mathcal{D}, \cdot, [\![\,]\!]^{\mathcal{D}}, \Psi\rangle$ and a type environment $\mathcal{V}$ do agree with the natural type theories which validate axiom (Ω-η) only when $\Phi = \mathcal{D}$. In fact we have $[\![\Omega]\!]^{\mathcal{D}}_{\mathcal{V}} = \mathcal{D}$ and $[\![\Omega \to \Omega]\!]^{\mathcal{D}}_{\mathcal{V}} = \{d \in \Psi \mid d = [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho}$ for some $x, M, \rho\} = \Phi$. So for these type theories the F-semantics coincides with the simple semantics. Therefore it is interesting to consider only strict type theories or natural type theories such that $\Omega \not\sim_{\nabla} \Omega \to \Omega$.

The following definition singles out the type theories which do not validate axiom (Ω-η) but which are adequate for the F-semantics as proved in Theorem 6.

Definition 27. *A type theory Σ^{∇} is an* F-type theory *iff*

1. *either Σ^{∇} is a strict type theory such that $\nu \notin \mathsf{C}^{\nabla}$ and for all $\psi \in \mathsf{C}^{\nabla}$, $A, B \in \mathsf{T}^{\nabla}$, there are $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that $\psi \cap (A \to B) \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$;*
2. *or Σ^{∇} is a strict type theory such that $\nu \in \mathsf{C}^{\nabla}$ and for all $\psi \in \mathsf{C}^{\nabla}$ either $\nu \leq_{\nabla} \psi$ or there are $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that $\psi \cap \nu \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$;*
3. *or Σ^{∇} is a natural type theory such that $\Omega \not\sim_{\nabla} \Omega \to \Omega$ and for all $\psi \in \mathsf{C}^{\nabla}$ there are $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that $\psi \cap (\Omega \to \Omega) \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$.* □

Notice that, if $\nu \in \mathsf{C}^{\nabla}$, condition (2) of the above definition implies that for all $\psi \in \mathsf{C}^{\nabla}$, $A, B \in \mathsf{T}^{\nabla}$, there are $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that $\psi \cap (A \to B) \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$.

Similarly, if $\Omega \in \mathsf{C}^{\nabla}$, condition (3) of the above definition implies that for all $\psi \in \mathsf{C}^{\nabla}$, $A, B \in \mathsf{T}^{\nabla}$, there are $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that $\psi \cap (A \to B) \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$.

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

Definition 28. *We define the predicate* fun *on* T^{∇} *by induction on the structure of types:*

1. $\mathsf{fun}(\psi) = \psi \sim_{\nabla} \nu$ or $\psi \sim_{\nabla} \bigcap_{i\in I}(A_i \to B_i)$ *for some* $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$;
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-type theories.

Proposition 6. *If Σ^{∇} is an F-type theory then* $\mathsf{fun}(A)$ *iff either $A \sim_{\nabla} \nu$ or $A \sim_{\nabla} \bigcap_{i\in I}(B_i \to C_i)$ for some $I \neq \emptyset, B_i, C_i \in \mathsf{T}^{\nabla}$.*□

To prove adequacy we will again use the filter structure $\langle \mathcal{F}^{\nabla}, F^{\nabla}, G^{\nabla}\rangle$, but with different type interpretation domains and type environments.

Definition 29. *Let Σ^{∇} be an F-type theory. Let:*

1. *Φ^{∇}_F be the functionality set defined by*

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

2. *$\mathcal{I}^{\nabla}_F$ be the type interpretation domain $\langle \mathcal{F}^{\nabla}, \cdot, [\![\,]\!]^{\nabla}, \Phi^{\nabla}_F\rangle$;*

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

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

4. $[\![\,]\!]_F^{\nabla} : \mathsf{T}^{\nabla} \to \mathsf{P}(\mathcal{F}^{\nabla})$ *be the mapping* $[\![\,]\!]_{\mathcal{V}_F^{\nabla}}^{\mathcal{I}_F^{\nabla}}$. □

When restricting to F-type theories, all filters which contain a functional type belong to the functionality set. As a consequence we get $\mathcal{V}_F^{\nabla}(\nu) = \{X \in \mathcal{F}^{\nabla} \mid \nu \in X\} = \Phi_F^{\nabla}$.

Lemma 2. *Let* Σ^{∇} *be an F-type theory and* $X \in \mathcal{F}^{\nabla}$. *Then* $A \in X$ *and* $\mathsf{fun}(A)$ *imply* $X \in \Phi_F^{\nabla}$.

Proof. We show that under the given conditions

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

Proof of $X \subseteq [\![\lambda y.xy]\!]_{\rho_0}^{\nabla}$. Take an arbitrary $B \in X$. Notice that if $\nu \in \mathsf{C}^{\nabla}$ then $\mathsf{fun}(A)$ implies $A \leq_{\nabla} \nu$ by Proposition 6. Moreover $\mathsf{fun}(A)$ implies $\mathsf{fun}(A \cap B)$ by Definition 28. Then either $A \cap B \sim_{\nabla} \nu$ or $A \cap B \sim_{\nabla} \bigcap_{i \in I}(C_i \to D_i)$ for some $I \neq \emptyset, C_i, D_i \in \mathsf{T}^{\nabla}$ again by Proposition 6. In the first case we get $\nu \leq_{\nabla} B$ and then $\vdash^{\nabla} \lambda y.xy : B$ by axiom (Ax-ν) and rule ($\leq_{\nabla}$). In the second case we can derive $x{:}\bigcap_{i \in I}(C_i \to D_i) \vdash^{\nabla} \lambda y.xy : \bigcap_{i \in I}(C_i \to D_i)$ using axiom (Ax) and rules ($\leq_{\nabla}$), ($\to$ E), ($\to$ I), and ($\cap$I). This implies $x{:}A \cap B \vdash^{\nabla} \lambda y.xy : B$ by rules ($\leq_{\nabla}$) and ($\leq_{\nabla}$ L). In both cases we conclude $B \in [\![\lambda y.xy]\!]_{\rho_0}^{\nabla}$ since $\emptyset \models \rho_0$ and $\{x{:}A \cap B\} \models \rho_0$.
Proof of $[\![\lambda y.xy]\!]_{\rho_0}^{\nabla} \subseteq X$. By Theorem 2, $B \in [\![\lambda y.xy]\!]_{\rho_0}^{\nabla}$ implies $x{:}C \vdash^{\nabla} \lambda y.xy : B$ for some $C \in X$. If $\Omega \in \mathsf{C}^{\nabla}$ and $B \sim_{\nabla} \Omega$ then $B \in X$ for all X. If $\nu \in \mathsf{C}^{\nabla}$ and $\nu \leq_{\nabla} B$ then $B \in X$ since $A \leq_{\nabla} \nu$ by Proposition 6. Otherwise we get $x{:}C, y{:}D_i \vdash^{\nabla} xy : E_i$ for some $I \neq \emptyset, D_i, E_i \in \mathsf{T}^{\nabla}$ such that $\bigcap_{i \in I}(D_i \to E_i) \leq_{\nabla} B$ by Theorem 1(4). This implies $x{:}C, y{:}D_i \vdash^{\nabla} x : F_i \to E_i$, $x{:}C, y{:}D_i \vdash^{\nabla} y : F_i$ for some $F_i \in \mathsf{T}^{\nabla}$ by Theorem 1(3). Using Theorem 1(1) we have $C \leq_{\nabla} F_i \to E_i$ and $D_i \leq_{\nabla} F_i$ for all $i \in I$, so we can conclude $C \leq_{\nabla} B$, i.e. $B \in X$, by rule (η). □

Theorem 6. *(Adequacy for the F-semantics)* $\Gamma \models_F^{\nabla} M : A$ *implies* $\Gamma \vdash^{\nabla} M : A$ *iff either* Σ^{∇} *is an F-type theory or* Σ^{∇} *is a natural type theory which validates axiom (*Ω*-*η*).*

Proof. The only interesting case is when Σ^{∇} is an F-type theory. First we check that the given conditions are necessary. Let $[\![x]\!]_{\rho}^{\mathcal{D}} \in [\![\psi \cap (A \to B)]\!]_{\mathcal{V}}^{\mathcal{D}}$ for some $\mathcal{D}, \mathcal{V}, \rho$. Then $[\![x]\!]_{\rho}^{\mathcal{D}} \in \{d \in \mathcal{D} \mid d = [\![\lambda x.M]\!]_{\rho}^{\mathcal{D}} \text{ for some } x, M, \rho\}$ since $[\![A \to B]\!]_{\mathcal{V}}^{\mathcal{D}} \in \{d \in \mathcal{D} \mid d = [\![\lambda x.M]\!]_{\rho}^{\mathcal{D}} \text{ for some } x, M, \rho\}$. This implies $[\![x]\!]_{\rho}^{\mathcal{D}} = [\![\lambda y.xy]\!]_{\rho}^{\mathcal{D}}$, so in particular we get $x{:}\psi \cap (A \to B) \models_F^{\nabla} \lambda y.xy : \psi \cap (A \to B)$. By a similar argument we can obtain $x{:}\psi \cap \nu \models_F^{\nabla} \lambda y.xy : \psi \cap \nu$ when $\nu \in \mathsf{C}^{\nabla}$. Therefore we have F-adequacy of $\lambda\cap^{\nabla}$ only if we can prove $x : \psi \cap (A \to B) \vdash^{\nabla} \lambda y.xy : \psi \cap (A \to B)$ (respectively $x : \psi \cap \nu \vdash^{\nabla} \lambda y.xy : \psi \cap \nu$ when $\nu \in \mathsf{C}^{\nabla}$). Let $\nu \notin \mathsf{C}^{\nabla}$.

$x{:}\psi \cap (A \to B) \vdash^{\nabla} \lambda y.xy : \psi \cap (A \to B) \Rightarrow x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\nabla} xy : B_i$
for some $I \neq \emptyset, A_i, B_i \in \mathsf{T}^{\nabla}$ such that
$\bigcap_{i \in I}(A_i \to B_i) \leq_{\nabla} \psi \cap (A \to B)$, by Theorem 1(4)
$\Rightarrow x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\nabla} x : D_i \to B_i$
and $x{:}\psi \cap (A \to B), y{:}A_i \vdash^{\nabla} y : D_i$
for some $D_i \in \mathsf{T}^{\nabla}$, by Theorem 1(3)
$\Rightarrow \psi \cap (A \to B) \leq_{\nabla} \bigcap_{i \in I}(D_i \to B_i)$ and $A_i \leq_{\nabla} D_i$
by Theorem 1(1)
$\Rightarrow \psi \cap (A \to B) \leq_{\nabla} \bigcap_{i \in I}(A_i \to B_i)$
by rule (η).

So we conclude $\psi \cap (A \to B) \sim_{\nabla} \bigcap_{i \in I}(A_i \to B_i)$.
Similarly from $x{:}\psi \cap \nu \models_F^{\nabla} \lambda y.xy : \psi \cap \nu$ we can show that either $\nu \leq_{\nabla} \psi$ or $\psi \cap \nu \sim_{\nabla} \bigcap_{i \in I}(A_i \to B_i)$ when $\nu \in \mathsf{C}^{\nabla}$.
For the vice versa we can prove that $[\![A]\!]_F^{\nabla} = \{X \in \mathcal{F}^{\nabla} \mid A \in X\}$ similarly to the proof of Lemma 1 using Lemma 2. Finally we conclude mimicking the proof of Theorem 4. □

The theories $\mathcal{AO}$ and $\mathcal{BCD}$, as well as the type theories of [11, 16, 21, 19, 8, 17] are adequate for the F-semantics. Moreover for the last five and $\mathcal{BCD}$ the simple semantics coincides with the F-semantics. The theory $\mathcal{B}a$ is an example of a theory 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 essentially in the interpretation of the *arrow* type constructor. In what follows we shall mainly follow the terminology of [14].

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

Following [24], 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* [23], 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 [18] 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* [5], [10] and to *polymorphic types* [18].

The crucial question in the semantics of types is the completeness of type assignment systems. Hindley proved in [14] that Curry's type assignment system is complete for all the mentioned semantics. More specifically [14] and [15] show the completeness for the simple semantics and moreover that:

1. $\Gamma \models^{\nabla}_{F} M : A$ if and only if $\Gamma \models^{\nabla}_{s} M : A$, 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 [5, 13, 9, 8, 3].

As far as the completeness with respect to the F-semantics of intersection-type assignment systems, over λ-models, we can cite [10],[27],[2]. In [10] the intersection-type assignment systems which are complete for the F-semantics are characterized. More specifically it is shown that an intersection-type assignment system is complete for the F-semantics if and only if types are invariant under β-equality of subjects and moreover the following rule (due to R.Hindley):

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

where A is an arbitrary type and Ω is the universal type, is a derived rule.

Yokouchi in [27] shows that if we add two suitable rules (quite similar to Hindley rule) to the intersection-type assignment system of [7] we obtain completeness for the F-semantics.

Abramsky and Ong, in [2], prove the completeness of the theory $\mathcal{AO}$, 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 theory which is adequate for the F-semantics, then Hindley's rule is admissible in $\lambda\cap^{\nabla}_{\Omega}$. This follows from the observation that under the given conditions for all $n \geq 0$ and for all $\psi \in \mathsf{C}^{\nabla}$ we can find $I \neq \emptyset, A_i^{(1)}, \ldots, A_i^{(n)}, B_i$ such that

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

We could not have used the syntactic approach based on term models introduced in [13] 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 MI reduces to an abstraction then a fortiori $M1$ reduces to an abstraction. Therefore $[\![x1]\!]_\rho$ is the representative of a function whenever $[\![xI]\!]_\rho$ is the representative of a function. Now consider the F-type theory $(\{\phi, \Omega\}, \mathcal{DM})$ where $\mathcal{DM} = \mathcal{AO} \cup \{\phi \cap (\Omega \to \Omega) \sim \Omega \to \phi\}$ [10]. We have $x{:}(\phi \to \phi) \to \Omega \to \Omega \vdash^{\mathcal{DM}}_{\Omega} xI : \Omega \to \Omega$ which implies, by soundness, $x{:}(\phi \to \phi) \to \Omega \to \Omega \models^{\mathcal{DM}}_{F} xI : \Omega \to \Omega$. By the above we get $x{:}(\phi \to \phi) \to \Omega \to \Omega \models^{\mathcal{DM}}_{F} x1 : \Omega \to \Omega$, but it is easy to check, using the Generation Lemma, that we cannot deduce $x1 : \Omega \to \Omega$ from $x{:}(\phi \to \phi) \to \Omega \to \Omega$.

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

References

1. S. Abramsky [1991]. Domain theory in logical form, *Ann. Pure Appl. Logic* **51**(1-2), pp. 1–77.
2. S. Abramsky and C.-H.L. Ong [1993]. Full abstraction in the lazy lambda calculus, *Inform. and Comput.* **105**(2), pp. 159–267.
3. S. van Bakel [1992]. Complete restrictions of the intersection type discipline, *Theoret. Comput. Sci.* **102**(1), pp. 135–163.
4. H. Barendregt *et al* [2000]. *Typed λ-calculus and applications*, North Holland (to appear).
5. H. Barendregt, M. Coppo and M. Dezani-Ciancaglini [1983]. A filter lambda model and the completeness of type assignment, *J. Symbolic Logic* **48**(4), pp. 931–940.
6. M. Coppo and M. Dezani-Ciancaglini [1980]. An extension of the basic functionality theory for the λ-calculus, *Notre Dame J. Formal Logic* **21**(4), pp. 685–693.
7. M. Coppo, M. Dezani-Ciancaglini and B. Venneri [1981]. Functional characters of solvable terms, *Z. Math. Logik Grundlag. Math.* **27**(1), pp. 45–58.
8. M. Coppo, M. Dezani-Ciancaglini and M. Zacchi [1987]. Type theories, normal forms, and D_∞-lambda-models, *Inform. and Comput.* **72**(2), pp. 85–116.
9. M. Coppo, F. Honsell, M. Dezani-Ciancaglini and G. Longo [1984]. Extended type structures and filter lambda models, *Logic colloquium '82 (Florence, 1982)*, North-Holland, Amsterdam, pp. 241–262.
10. M. Dezani-Ciancaglini and I. Margaria [1986]. A characterization of F-complete type assignments, *Theoret. Comput. Sci.* **45**(2), pp. 121–157.
11. L. Egidi, F. Honsell and S. Ronchi Della Rocca [1992]. Operational, denotational and logical descriptions: a case study, *Fund. Inform.* **16**(2), pp. 149–169.
12. E. Engeler [1981]. Algebras and combinators, *Algebra Universalis* **13**(3), pp. 389–392.
13. J.R. Hindley, [1982]. The simple semantics for Coppo–Dezani–Sallé types, *International symposium on programming (Turin, 1982)*, Springer, Berlin, pp. 212–226.
14. J.R. Hindley, [1983]. The completeness theorem for typing λ-terms, *Theoret. Comput. Sci.* **22**, pp. 1–17.
15. J.R. Hindley, [1983]. Curry's types are complete with respect to the F-semantics too, *Theoret. Comput. Sci.* **22**, pp. 127–133.
16. F. Honsell and M. Lenisa [1999]. Semantical analysis of perpetual strategies in λ-calculus, *Theoret. Comput. Sci.* **212**(1-2), pp. 183–209. Gentzen (Rome, 1996).
17. F. Honsell and S. Ronchi Della Rocca [1992]. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus, *J. Comput. System Sci.* **45**(1), pp. 49–75.
18. J.Mitchell [1988]. Polymorphic type inference and containment, *Inform. and Comput.* **76**, pp. 211–249.
19. D. Park [1976]. The Y-combinator in Scott's λ-calculus models (revised version), Theory of Computation Report 13, Department of Computer Science, University of Warwick.

20. G.D. Plotkin [1993]. Set-theoretical and other elementary models of the λ-calculus, *Theoret. Comput. Sci.* **121**(1-2), pp. 351–409.
21. D.S. Scott [1972]. Continuous lattices, *Toposes, algebraic geometry and logic*, Springer, Berlin, pp. 97–136. Lecture Notes in Math., Vol. 274.
22. D.S. Scott [1975]. Open problem, *Lambda Calculus and Computer Science Theory*, Lecture Notes in Computer Science 37, Springer, Berlin, p. 369.
23. D.S. Scott [1976]. Data types as lattices, *SIAM J. Comput.* **5**(3), pp. 522–587.
Semantics and correctness of programs.
24. D.S. Scott [1980]. Lambda Calculus: some models, some philosophy *The Kleene Symposium*, North-Holland, Amsterdam, pp. 223–266.
25. D.S. Scott [1980]. Letter to Albert Meyer.
26. S. Vickers [1989]. *Topology via logic*, Cambridge University Press, Cambridge.
27. H. Yokouchi [1994]. F-semantics for type assignment systems, *Theoret. Comput. Sci.* **129**, pp. 39-77.