Semantic Techniques for Deriving Coinductive Characterizations of Observational Equivalences for λ-calculi*

Marina Lenisa

Dipartimento di Matematica e Informatica, Università di Udine, Italy.
lenisa@dimi.uniud.it

Abstract. Coinductive (applicative) characterizations of various observational congruences which arise in the semantics of λ-calculus, for various reduction strategies, are discussed. Two semantic techniques for establishing the coincidence of the applicative and the contextual equivalences are analyzed. The first is based on *intersection types*, the second is based on a *mixed induction-coinduction principle*.

Introduction

This paper is part of a general project on finding *elementary proof principles* for reasoning *rigorously* on possibly *infinite* objects [8, 13]. Often, in dealing with infinite or circular objects, pure structural induction can be applied only after cumbersome encodings. Thus, alternative reasoning principles are called for, such as *coinduction* and *mixed induction-coinduction principles* (see [16, 8]). In this paper, we focus on the behaviour of λ-terms when they are evaluated according to a given *reduction strategy*. In particular, we introduce two general semantic techniques for deriving the coincidence of the *applicative* equivalence with the *observational* (operational, contextual) equivalence for various reduction strategies. This amounts to a *coinductive* characterization of the observational equivalence and hence it gives rise immediately to a *coinduction principle*; the particular nature of one of the techniques gives rise also to a *mixed induction-coinduction principle*. This paper is a companion to [14], where syntactic techniques are discussed.

A reduction strategy is a procedure for determining, for each λ-term, a specific β-redex in it, to contract. Let $\Lambda(C)$ $(\Lambda^0(C))$ denote the set of (closed) λ-terms, where C is a set of basic constants. When $C = \emptyset$, we write Λ (Λ^0). A (possibly non-deterministic) strategy can be formalized as a relation $\rightarrow_\sigma \subseteq \Lambda(C) \times \Lambda(C)$ $(\Lambda^0(C) \times \Lambda^0(C))$ such that, if $(M, N) \in \rightarrow_\sigma$ (also written infix as $M \rightarrow_\sigma N$), then N is a possible result of applying $\rightarrow_\sigma$ to M. A particular set of terms which do not belong to the domain of $\rightarrow_\sigma$ are called *σ-values*, denoted by Val_σ. Given $\rightarrow_\sigma$, we can define the *evaluation relation* $\Downarrow_\sigma \subseteq \Lambda(C) \times \Lambda(C)$

* Work supported by CNR, EC HCM Project No. CHRX-CT92.0046 Lambda-Calcul Typé, and MURST 40% grant.

$(\Lambda^0(C) \times \Lambda^0(C))$, such that $M \Downarrow_\sigma N$ holds if and only if there exists a reduction path from M to the σ-value N. If there exists N such that $M \Downarrow_\sigma N$, then $\rightarrow_\sigma$ *halts successfully* (terminates) on M, otherwise it *diverges* on M ($M \not\Downarrow_\sigma$).

Each reduction strategy induces an *operational semantics*, in that we can imagine a machine which evaluates terms by implementing the given strategy. The *observational equivalence* arises if we consider programs as *black boxes* and only observe their "halting properties".

Definition 1 (σ-observational Equivalence). Let $\rightarrow_\sigma$ be a reduction strategy and let $M, N \in \Lambda^0(C)$. The *observational equivalence* $\approx_\sigma$ is defined by
$M \approx_\sigma N$ iff $\forall C[\;].(C[M], C[N] \in \Lambda^0(C) \Rightarrow (C[M] \Downarrow_\sigma \Leftrightarrow C[N] \Downarrow_\sigma))$.

Showing σ-observational equivalences by induction on computation steps is difficult. Powerful proof-principles, allowing to factorize this task, are precious. Coinduction and mixed induction-coinduction principles for establishing $\approx_\sigma$ follow from the fact that $\approx_\sigma = \approx_\sigma^{app}$, where $\approx_\sigma^{app}$ denotes the *applicative equivalence* induced by $\rightarrow_\sigma$ (see Section 1).

In this paper, we introduce two *general* methods for showing $\approx_\sigma = \approx_\sigma^{app}$:
1. *domain logic* method based on the *intersection types* presentation of a suitable computationally adequate *CPO*-λ-model;
2. *logical relations* method based on a mixed induction-coinduction principle.

Both methods are semantical, in the sense that they are based on a suitable *computationally adequate CPO*-λ-model. The "domain logic method" is the generalization of the method originally used by Abramsky and Ong in [1] for the special case of the lazy reduction strategy. The "logical relations method" is the generalization of the method originally used by Pitts in [16, 17, 18] for the special cases of by-name and by-value lazy reduction strategies. Pitts' technique is based on the *minimal invariance property* of the model, as is the case for initial models. In this paper we show how to extend it to many more strategies and models. Both methods presented in this paper can be used to show that $\approx_\sigma = \approx_\sigma^{app}$ for all the reduction strategies in [8], thus solving some open questions raised there. The strategies in [8] are:
• $\rightarrow_l \subseteq \Lambda^0 \times \Lambda^0$, the *lazy call-by-name* strategy, which reduces the leftmost β-redex not appearing within a λ-abstraction (see [1]);
• $\rightarrow_v \subseteq \Lambda^0 \times \Lambda^0$, the *lazy call-by-value* strategy ([19]), which reduces the *leftmost* β-redex, not within an abstraction, whose argument is an abstraction;
• $\rightarrow_o \subseteq \Lambda^0(\{\Omega\}) \times \Lambda^0(\{\Omega\})$, a non-deterministic strategy, which rewrites λ-terms containing occurrences of the constant Ω by reducing any β-redex. The congruence $\approx_o$ cannot be realized by a continuously complete CPO model ([10]);
• $\rightarrow_h \subseteq \Lambda \times \Lambda$, the *head call-by-name* strategy, which reduces the leftmost β-redex, if the term is not in head normal form;
• $\rightarrow_n \subseteq \Lambda \times \Lambda$, the leftmost$\beta$-*normalizing* strategy; $\approx_n$ is modeled in [4];
• $\rightarrow_p \subseteq \Lambda \times \Lambda$, Barendregt's *perpetual* strategy, which reduces the leftmost β-redex not in the operator of a β-redex, which is either an I-redex, or a K-redex whose argument is a normal form; $\rightarrow_p$ terminates exactly on *strongly normalizing* terms; the model in [7] for the λN^0-calculus is computationally adequate w.r.t. $\approx_p$ ([9]).

In this paper, we outline the two methods in full generality, but, for lack of space, we give details only for $\to_v$ and $\to_h$. These are representative of the difficulties one has to deal with in the generalization of [1] and [16, 17, 18]. The latter is a prototype of a strategy which has to take into account also *open* terms and does not have an "appropriate" initial model.

In the paper we use λ-calculus concepts and notation as defined in [2, 8]. The paper is organized as follows. In Section 1 we introduce the problem of characterizing coinductively contextual equivalences via applicative equivalences. In Section 2 we define $\to_v$ and $\to_h$. In Section 3 we give a description via intersection types of the computationally adequate models for $\approx_h$ and $\approx_v$. In Section 4 we present in general the domain logic method and the logical relations method, and we carry out the details for the case of $\approx_v$ and $\approx_h$, respectively. Final remarks appear in Section 5.

The author is grateful to Mariangiola Dezani, Furio Honsell, and Andrew Pitts for useful discussions.

1 Coinductive Characterizations via Applicative Equivalences

Given a reduction strategy $\to_\sigma$, the σ*-applicative equivalence*, $\approx_\sigma^{app}$, differs from the observational one, in that the behaviour of the programs is tested only on applicative (closed) contexts.

Definition 2. Let $\approx_\sigma^{app} \subseteq \Lambda^0(C) \times \Lambda^0(C)$ be the *applicative equivalence*:
$M \approx_\sigma^{app} N \iff \forall P_1, \ldots, P_n \in \Lambda^0(C).\ (MP_1 \ldots P_n \Downarrow_\sigma \Leftrightarrow NP_1 \ldots P_n \Downarrow_\sigma)$.

The equivalence $\approx_\sigma^{app}$ has a coinductive characterization:

Lemma 1. *The applicative equivalence $\approx_\sigma^{app}$ can be viewed as the greatest fixed point of the monotone operator $\Psi_\sigma : \mathcal{P}(\Lambda^0(C) \times \Lambda^0(C)) \to \mathcal{P}(\Lambda^0(C) \times \Lambda^0(C))$*

$$\begin{aligned}\Psi_\sigma(R) = \{(M,N) \mid\ & (M \not\Downarrow_\sigma\ \wedge N \not\Downarrow_\sigma\ \wedge \forall P \in \Lambda^0(C).\ (MP, NP) \in R) \vee \\ & (M \Downarrow_\sigma\ \wedge N \Downarrow_\sigma\ \wedge \forall P \in \Lambda^0(C).\ (MP, NP) \in R)\}.\end{aligned}$$

An immediate consequence is the validity of the *coinduction principle*:

$$\frac{(M,N) \in R \quad R \text{ is a } \Psi_\sigma\text{-bisimulation}}{M \approx_\sigma^{app} N}\ ,$$

where a Ψ_σ*-bisimulation* is a relation $R \subseteq \Lambda^0(C) \times \Lambda^0(C)$ s.t. $R \subseteq \Psi_\sigma(R)$.

Remark. i) Another important relation on $\Lambda^0(C) \times \Lambda^0(C)$ is the *applicative pre-equivalence* $\leq_\sigma^{app}$ defined as follows:
$M \leq_\sigma^{app} N \iff \forall P_1, \ldots, P_n \in \Lambda^0(C).\ (MP_1 \ldots P_n \Downarrow_\sigma \Rightarrow NP_1 \ldots P_n \Downarrow_\sigma)$.
Then $\approx_\sigma^{app} = \leq_\sigma^{app} \cap (\leq_\sigma^{app})^{-1}$. Using a suitable monotone operator $\Psi_\sigma^{\leq}$, similar to that of Lemma 1, one can give a coinductive characterization also to $\leq_\sigma^{app}$.
ii) When $\Downarrow_\sigma$ is axiomatized on the whole Λ, we can define the *applicative equivalence extended to open contexts* $\approx_\sigma^{app,ext}$: let $M, N \in \Lambda^0(C)$,
$M \approx_\sigma^{app,ext} N \iff \forall P_1, \ldots, P_n \in \Lambda(C).\ (MP_1 \ldots P_n \Downarrow_\sigma \Leftrightarrow NP_1 \ldots P_n \Downarrow_\sigma)$.
$\approx_\sigma^{app,ext}$ will be useful in the sequel for reasoning on $\approx_\sigma^{app}$, since the two equivalences coincide in many interesting cases.

If $\approx_\sigma = \approx_\sigma^{app}$, then the coinduction principle above can be used to establish directly the observational equivalence. Hence the natural question arises: for which strategies σ's do the two equivalences coincide? Notice that there are σ's such that $\approx_\sigma \not\supseteq \approx_\sigma^{app}$:

Examples. 1) Let $Val_\tau = \{\lambda x.xM_1M_2 \mid M_1 =_\beta M_2\}$, where $=_\beta$ is β-conversion. Let $\rightarrow_\tau \subseteq \Lambda \times \Lambda$ be the reduction strategy which reduces the leftmost β-redex, if it exists, and if the term is not in Val_τ.
Then one can easily show that:
i) $M \approx_\tau N$ if and only if $M =_\beta N$, but
ii) let P, Q be two β-different fixed point of K ($K \equiv \lambda xy.x$), i.e. $\forall M.\ KPM =_\beta P$, $\forall M.\ KQM =_\beta Q$, but $P \neq_\beta Q$, then $P \approx_\tau^{app} Q$, but $P \not\approx_\tau Q$.
2) A closed term $\lambda x.M$ is said to be an "eraser" if $x \notin FV(M)$. The strategy $\rightarrow_e \subseteq \Lambda^0 \times \Lambda^0$ rewrites closed λ-terms not in normal form which are not erasers by reducing any β-redex. $Val_e = \{M \in \Lambda^0 \mid M \text{ is an eraser}\}$.
One can easily show that $\approx_e^{app} \neq \approx_e$. In fact, let e.g. $M \equiv \lambda x.(\Delta\Delta(xO))$, and $N \equiv \lambda x.(\Delta\Delta x)$, where $\Delta \equiv \lambda x.xx$ and $O \equiv \lambda x.I$. Then, it is easy to check that $M \approx_e^{app} N$, but if we take $C[\ ] \equiv \lambda y.([\ ]\lambda z.zy)$, then $C[M] \Downarrow_e$, while $C[N] \not\Downarrow_e$.
However, $\approx_e$ can be characterized coinductively since it coincides with $\approx_o^{app}$.

However, for many interesting strategies in the literature, one can show that $\approx_\sigma = \approx_\sigma^{app}$, see e.g. [1, 5, 8, 11, 17, 15]. The techniques introduced in this paper are rather general and both can be used for establishing the coincidence for all the strategies listed in the Introduction.

2 Two Interesting Strategies

In this section we briefly discuss the reduction strategies $\rightarrow_v$ and $\rightarrow_h$, which we shall focus on in Section 4. In particular we axiomatize the corresponding evaluation relations.

$\rightarrow_v$ **strategy.** The *lazy call-by-value* strategy $\rightarrow_v \subseteq \Lambda^0 \times \Lambda^0$ reduces the *leftmost* β-redex, not appearing within a λ-abstraction, whose argument is a λ-abstraction. $Val_v = \{\lambda x.M \mid M \in \Lambda\} \cap \Lambda^0$. The evaluation $\Downarrow_v$ is the least binary relation over $\Lambda^0 \times Val_v$ satisfying the following rules:

$$\frac{}{\lambda x.M \Downarrow_v \lambda x.M} \qquad\qquad \frac{M \Downarrow_v \lambda x.P \quad N \Downarrow_v Q \quad P[Q/x] \Downarrow_v U}{MN \Downarrow_v U}$$

The notion of β-reduction which is correct w.r.t. $\approx_v$[2] is the $\rightarrow_{\beta_v} \subseteq \Lambda \times \Lambda$ [19], i.e. $(\lambda x.M)N \rightarrow_{\beta_v} M[N/x]$, if N is a variable or an abstraction.
The initial solution in $CPO_\perp$, D^v, of the domain equation $D \simeq [D \rightarrow_\perp D]_\perp$, where $[D \rightarrow_\perp D]_\perp$ denotes the lifted space of strict Scott-continuous functions, gives a computationally adequate model for $\approx_v$. This model is studied in [5].

[2] The β-reduction $\rightarrow_{\beta_r}$ is correct w.r.t. $\approx_\sigma$ if, for all $M, N \in \Lambda^0(C)$, $M =_{\beta_r} N \Rightarrow M \approx_\sigma N$.

$\rightarrow_h$ **strategy.** The *head call-by-name* strategy $\rightarrow_h \subseteq \Lambda \times \Lambda$ reduces the *leftmost* β-redex, if the term is not in head normal form. Val_h is the set of λ-terms in head normal form. The evaluation $\Downarrow_h$ is the least binary relation over $\Lambda \times Val_h$ satisfying the following rules, for $n \geq 0$:

$$\frac{}{xM_1 \dots M_n \Downarrow_h xM_1 \dots M_n} \qquad \frac{M \Downarrow_h N}{\lambda x.M \Downarrow_h \lambda x.N} \qquad \frac{M[N/x]M_1 \dots M_n \Downarrow_h P}{(\lambda x.M)NM_1 \dots M_n \Downarrow_h P}$$

Classical β-reduction is correct w.r.t. $\approx_h$. The canonical D_∞ model of Scott is fully abstract w.r.t. $\approx_h$ (see [2]). We will refer to the model D_∞ as D^h.

3 Computationally Adequate Models

In this section we phrase in terms of intersection types ([3]) the computational adequacy of an algebraic lattice λ-model D^σ. We assume that the finitary logical presentation of D^σ is given by the intersection types theory $\mathcal{T}_\sigma = (T_\sigma, \leq_\sigma)$.

Definition 3. Let $\mathcal{T}_\sigma = (T_\sigma, \leq_\sigma)$ be the intersection type theory corresponding to D^σ. Then $\qquad (T_\sigma \ni)\ \phi ::= \ \alpha \ \mid \ \phi \wedge \phi \ \mid \ \phi \rightarrow \phi$,
where $\alpha \in TB_\sigma$, and TB_σ is a non-empty set of base types.

The relation $\leq_\sigma$ on types is the least relation containing the set IB_σ of inequalities involving the base types in TB_σ, and closed under the rules:

$$\phi \leq_\sigma \phi \qquad \phi_1 \wedge \phi_2 \leq_\sigma \phi_1 \qquad \phi_1 \wedge \phi_2 \leq_\sigma \phi_2$$

$$\frac{\phi_1 \leq_\sigma \phi_2 \quad \phi_2 \leq_\sigma \phi_3}{\phi_1 \leq_\sigma \phi_3} \qquad \frac{\phi \leq_\sigma \phi_1 \quad \phi \leq_\sigma \phi_2}{\phi \leq_\sigma \phi_1 \wedge \phi_2} \qquad \frac{\phi_2 \leq_\sigma \phi_1 \quad \psi_1 \leq_\sigma \psi_2}{\phi_1 \rightarrow \psi_1 \leq_\sigma \phi_2 \rightarrow \psi_2}$$

$$\phi \rightarrow \phi_1 \wedge \phi_2 \leq_\sigma (\phi \rightarrow \phi_1) \wedge (\phi \rightarrow \phi_2) \qquad (\phi \rightarrow \phi_1) \wedge (\phi \rightarrow \phi_2) \leq_\sigma \phi \rightarrow \phi_1 \wedge \phi_2$$

We put $\phi_1 =_\sigma \phi_2$ if and only if $\phi_1 \leq_\sigma \phi_2 \ \wedge \ \phi_2 \leq_\sigma \phi_1$. A *type environment* is a partial function $\Gamma : Vars \longrightarrow T_\sigma$. Let Env_Γ denote the set of type environments. The type assignment system S_σ includes the following rules and, possibly, a nonempty set SB_σ of axioms involving base types, and base constants of the language: $\qquad \Gamma[\phi/x] \vdash_\sigma x : \phi \quad var$

$$\frac{\Gamma \vdash_\sigma M : \phi \quad \Gamma \vdash_\sigma M : \psi}{\Gamma \vdash_\sigma M : \phi \wedge \psi} \ \wedge I \qquad \frac{\Gamma \leq_\sigma \Delta \quad \Delta \vdash_\sigma M : \phi \quad \phi \leq_\sigma \psi}{\Gamma \vdash_\sigma M : \psi} \ \leq$$

$$\frac{\Gamma[\phi/x] \vdash_\sigma M : \psi}{\Gamma \vdash_\sigma \lambda x.M : \phi \rightarrow \psi} \ \rightarrow I \qquad \frac{\Gamma \vdash_\sigma M : \phi \rightarrow \psi \quad \Gamma \vdash_\sigma N : \phi}{\Gamma \vdash_\sigma MN : \psi} \ \rightarrow E$$

The canonical interpretation $[\![\ ]\!]^{D^\sigma}$ in the filter model D^σ can be characterized as follows: $[\![M]\!]^{D^\sigma}_\rho = \{\phi \mid \exists \Gamma \sqsubseteq_{D^\sigma} \rho.\ \Gamma \vdash_\sigma M : \phi\}$, where $\rho : Var \rightarrow VD^\sigma$, for a suitable $VD^\sigma \subseteq D^\sigma$. $\Gamma \sqsubseteq_{D^\sigma} \rho$ means that $\forall x \in Var.\ \Gamma(x) \sqsubseteq_{D^\sigma} \rho(x)$, and $\sqsubseteq_{D^\sigma}$ denotes the order relation on D^σ.

The computational adequacy of the model D^σ (i.e. for $M, N \in \Lambda^0(C)$, $[\![M]\!]^{D^\sigma} \sqsubseteq_{D^\sigma} [\![N]\!]^{D^\sigma} \;\Rightarrow\; M \leq_\sigma N$) can now take the strong form:

Theorem 2 (Computational Adequacy). *There exists a filter of types $T_\sigma^{conv} \subseteq T_\sigma$, and a subset Env_Γ^{conv} of $\mathcal{T}_\sigma^{conv}$-type environments such that*
$$M \Downarrow_\sigma \iff \exists \Gamma \in Env_\Gamma^{conv}.\ \exists \phi \in\ T_\sigma^{conv}.\ (\Gamma \vdash_\sigma M : \phi).$$

Intersection Type Description of the Model D^v. We recall the intersection type description of the model D^v of [5].

Definition 4. Let $\mathcal{T}_v = (T_v, \leq_v)$ be the intersection type theory:
$$(T_v \ni)\ \phi ::=\ \nu \ \mid\ \phi \wedge \phi\ \mid\ \phi \to \phi.$$
The axiomatization of $\leq_v$ consists of the standard rules of Definition 3, and of the following rule involving the base type ν: $\quad \phi \leq_v \nu$.

The type system S_v consists of the rules of Definition 3, and moreover:
$$\Gamma \vdash_v \lambda x.M : \nu \quad\ \nu\ .$$

Theorem 3 (Comp. Ad. [5]). $M \Downarrow_v \iff \exists \phi \in\ T_v.\ \vdash_v M : \phi$.

Intersection Type Description of the Model D^h. An intersection type description of the model D^h is:

Definition 5. Let $\mathcal{T}_h = (T_h, \leq_h)$ be the intersection type theory:
$$(T_h \ni)\ \phi ::=\ \omega\ \mid\ 1\ \mid\ \phi \wedge \phi\ \mid\ \phi \to \phi.$$
The axiomatization of $\leq_h$ consists of the standard rules of Definition 3, and moreover of the following rules involving the base types ω and 1:
$\phi \leq_h \omega \quad\ \omega \leq_h \omega \to \omega \quad\ 1 =_h \omega \to 1.$

The type system S_h consists of the rules of Definition 3 and moreover:
$$\Gamma \vdash_h M : \omega \quad\ \omega\ .$$

Theorem 4 (Comp. Ad.). $M \Downarrow_h \iff \exists \Gamma\ \exists \phi \neq_h \omega.\ \Gamma \vdash_h M : \phi$.

4 Two Techniques for Showing $\approx_\sigma^{app} = \approx_\sigma$

In this section we analyze two general techniques for establishing $\approx_\sigma^{app} = \approx_\sigma$. The inclusion $\approx_\sigma \subseteq \approx_\sigma^{app}$ is immediate; the other inclusion is in general difficult to prove. The techniques we present are:
i) *Domain logic* method based on the *intersection types* presentation of a *computationally adequate* model;
ii) *Logical relations* method based on an induction-coinduction principle.

Both techniques are based on the existence of a suitable computationally adequate CPO-model. A special case of the former was described in the case of $\to_l$ by Abramsky and Ong in [1]. A version of the latter, applicable only to computationally adequate CPO-models which satisfy a "minimal invariance property", appears in [16, 17, 18]. We generalize and streamline both techniques.

Both methods are used to show that $\approx_\sigma^{app}$ is a congruence w.r.t. application. In fact, in order to prove that $\approx_\sigma^{app} \subseteq \approx_\sigma$, it is sufficient to show that (Theorem 8):
1. $\approx_\sigma^{app}$ is a congruence w.r.t. application, i.e. for all $M, N, P, Q \in \Lambda^0(C)$,

$$M \approx_\sigma^{app} N \ \wedge \ P \approx_\sigma^{app} Q \implies MP \approx_\sigma^{app} NQ \ ;$$

2. $\approx_\sigma^{app}$ is a congruence w.r.t. λ-abstraction, i.e., $\forall M, N \in \Lambda(C)$ such that $FV(M, N) \subseteq \{x_1, \ldots, x_n\}$, $\forall P_1, \ldots, P_n \in \Lambda^0(C)$. $M[P_i/x_i] \approx_\sigma^{app} N[P_i/x_i] \implies$

$$\lambda x_1 \ldots x_n.M \approx_\sigma^{app} \lambda x_1 \ldots x_n.N \ .$$

(In case the strategy is by-value, i.e. for $\sigma = v, p$, $P_1, \ldots, P_n$ are chosen to be convergent terms.)

The congruence of $\approx_\sigma^{app}$ w.r.t. application is in general much more difficult to show than that w.r.t. λ-abstraction. The latter follows immediately from the extensionality of $\approx_\sigma^{app}$ (Theorem 5). The proof of Theorem 5 is problematic only for $\sigma = n$, where one needs to exploit extensively the separability technique of [12]. For lack of space, we omit this proof.

Theorem 5 (Extensionality of $\approx_\sigma^{app}$). *i) Let $\sigma = l, v$. Let $M, N \in \Lambda^0$ be such that $M \Downarrow_\sigma$, $N \Downarrow_\sigma$. If, for all $P \in \Lambda^0$, $MP \approx_\sigma^{app} NP$, then $M \approx_\sigma^{app} N$.*
ii) Let $\sigma = o, h, p, n$. Let $M, N \in \Lambda^0(C)$. If, for all $P \in \Lambda^0(C)$, $MP \approx_\sigma^{app} NP$, then $M \approx_\sigma^{app} N$.

Theorem 6. *$\approx_\sigma^{app}$ is a congruence w.r.t. λ-abstraction, for $\sigma \in \{l, v, o, h, n, p\}$.*

Proof. We show that, for $M, N \in \Lambda$ s.t. $FV(M, N) \subseteq \{x\}$,
$\forall P \in \Lambda^0$*(possibly convergent)*. $M[P/x] \approx_\sigma^{app} N[P/x] \ \Rightarrow \ \lambda x.M \approx_\sigma^{app} \lambda x.N$.
For $\sigma = l, v$ this is immediate. For $\sigma = o, h, n$ the proof follows from the Extensionality Theorem, using the fact that $(\lambda x.M)P \approx_\sigma^{app} M[P/x]$, which in turn follows from the correctness of the β-reduction w.r.t. $\approx_\sigma^{app}$. For $\sigma = p$, the proof follows from the fact that, for all $M \in \Lambda$, $(\exists P \in \Lambda^0.\ M[P/x] \Downarrow_p) \iff M \Downarrow_p$. The implication $(\Rightarrow)$ in this latter fact follows since $\rightarrow_p$ is perpetual. The other implication is proved by computation induction, choosing as P a suitable permutator. $\square$

Corollary 7. *$\approx_\sigma^{app,ext} = \approx_\sigma^{app}$, for $\sigma = h, n, p$.*

Theorem 8. *Suppose that $\approx_\sigma^{app}$ is a congruence w.r.t. λ-abstraction and application. Then $\approx_\sigma^{app} \subseteq \approx_\sigma$.*

Proof. We prove by induction on the context $C[\,]$ that:
$M \approx_\sigma^{app} N \implies \forall C[\,].(C[M], C[N] \in \Lambda(C) \wedge FV(C[M], C[N]) \subseteq \{x_1, \ldots, x_n\} \Rightarrow$
$\forall P_1 \ldots P_n \in \Lambda^0(C).\ C[M][P_i/x_i] \approx_\sigma^{app} C[N][P_i/x_i])$. $\square$

4.1 Domain Logic Method

This is a semantical method for showing $\approx_\sigma^{app} \subseteq \approx_\sigma$, based on the logical description via intersection types of a computationally adequate CPO-model. This method can be also used to derive the computational adequacy of D^σ, for

$\sigma = l, v, o, p$. However, for $\sigma = h, n$, the computational adequacy must be assumed at the outset. It generalizes that in [1] in two main respects:
i) we relax the condition on type interpretation on applicative structures, by requiring only that the type interpretation be *adequate*. In fact, in order to show the crucial Theorem 15, a uniformly defined notion of type interpretation on the whole class of applicative structures (like in [1]) is not necessary. An *adequate* type interpretation on the applicative structure $\mathcal{A}$ is enough;
ii) we use, in place of λ-model, the weaker notion of σ-combinatory algebra.

In order to state the crucial result of this method, i.e. Theorem 15, we need a number of results concerning the notions of *applicative σ-structure with convergence* (Definition 6) and of *combinatory σ-algebra* (Definition 11).

Definition 6 (Applicative σ-structure with Convergence). An *applicative σ-structure with convergence* is a structure $\mathcal{A} = (A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}})$ endowed with a notion of applicative equivalence $\approx^{app}_{\mathcal{A}} \subseteq A \times A$ such that
• $\{a \in A \mid a \Downarrow_{\mathcal{A}}\} \neq \emptyset, A$;
• if the application between λ-terms is right (left) strict w.r.t. $\Downarrow_{\sigma}$, then also $\bullet_{\mathcal{A}}$ is right (left) strict w.r.t. $\Downarrow_{\mathcal{A}}$;
• $a \approx^{app}_{\mathcal{A}} b \iff \forall k \geq 0\ \forall c_1, \ldots, c_k \in A\ (\forall i.\ c_i \Downarrow_{\mathcal{A}} \Rightarrow (ac_1 \ldots c_k \Downarrow_{\mathcal{A}} \Leftrightarrow bc_1 \ldots c_k \Downarrow_{\mathcal{A}}))$, if the application between λ-terms is right strict w.r.t. $\Downarrow_{\sigma}$,
$a \approx^{app}_{\mathcal{A}} b \iff \forall k \geq 0\ \forall c_1, \ldots, c_k \in A\ (ac_1 \ldots c_k \Downarrow_{\mathcal{A}} \Leftrightarrow bc_1 \ldots c_k \Downarrow_{\mathcal{A}})$, otherwise.

The model D^{σ} can be viewed as an applicative σ-structure whose notion of convergence is: $a \Downarrow_{D^{\sigma}} \iff \exists \phi \in T'_{\sigma}.\ a \sqsupseteq_{D^{\sigma}} \phi$, for any filter $T'_{\sigma} \subseteq T_{\sigma}$. Throughout this section we shall focus on the filter T^{conv}_{σ} of Theorem 2. This is not strictly necessary until Theorem 15, below. Notice that until then no mention is made of the fact that D^{σ} is computationally adequate w.r.t. precisely that T^{conv}_{σ}; and even then this assumption is not necessary in all cases.

We can define various notions of interpretations for the types in $\mathcal{T}_{\sigma}$ on the class of applicative σ-structures with convergence. In the following definition, we isolate the class of type interpretations we are interested in:

Definition 7. Let $\mathcal{A} = (A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}})$ be an applicative σ-structure with convergence. An interpretation of the type theory $\mathcal{T}_{\sigma}$, $[\![\ ]\!]^{t\mathcal{A}} : T_{\sigma} \to \mathcal{P}(A)$, is *adequate* on $\mathcal{A}$, if $[\![\ ]\!]^{t\mathcal{A}}$ satisfies the following properties:

– $[\![\phi \wedge \psi]\!]^{t\mathcal{A}} = [\![\phi]\!]^{t\mathcal{A}} \cap [\![\psi]\!]^{t\mathcal{A}}$

$$[\![\phi \to \psi]\!]^{t\mathcal{A}} =$$
$$= \begin{cases} \{a \in A \mid a \Downarrow_{\mathcal{A}}\ \wedge \forall c \in [\![\phi]\!]^{t\mathcal{A}}.\ (c \Downarrow_{\mathcal{A}} \Rightarrow\ a \bullet_{\mathcal{A}} c \in [\![\psi]\!]^{t\mathcal{A}})\} & \text{if } (1) \\ \{a \in A \mid a \Downarrow_{\mathcal{A}}\ \wedge \forall c \in [\![\phi]\!]^{t\mathcal{A}}.\ a \bullet_{\mathcal{A}} c \in [\![\psi]\!]^{t\mathcal{A}}\} & \text{if } (2) \\ \{a \in A \mid \forall c \in [\![\phi]\!]^{t\mathcal{A}}.\ (c \Downarrow_{\mathcal{A}} \Rightarrow\ a \bullet_{\mathcal{A}} c \in [\![\psi]\!]^{t\mathcal{A}})\} & \text{if } (3) \\ \{a \in A \mid \forall c \in [\![\phi]\!]^{t\mathcal{A}}.\ a \bullet_{\mathcal{A}} c \in [\![\psi]\!]^{t\mathcal{A}}\} & \text{otherwise} \end{cases}$$

where: (1) $\bullet_{\mathcal{A}}$ is left and right strict w.r.t. $\Downarrow_{\mathcal{A}}$;
(2) $\bullet_{\mathcal{A}}$ is left strict w.r.t. $\Downarrow_{\mathcal{A}}$; (3) $\bullet_{\mathcal{A}}$ is right strict w.r.t. $\Downarrow_{\mathcal{A}}$.

- $\phi \leq_\sigma \psi \;\Rightarrow\; [\![\phi]\!]^{t\mathcal{A}} \subseteq [\![\psi]\!]^{t\mathcal{A}}$
- $a \Downarrow_\mathcal{A} \iff \exists \phi \in T_\sigma^{conv}.\ a \in [\![\phi]\!]^{t\mathcal{A}}$.

Proposition 9. *The type interpretation $[\![\ ]\!]^{tD^\sigma}$ on D^σ, defined as, $a \in [\![\phi]\!]^{tD^\sigma}$ if and only if $a \sqsupseteq_{D^\sigma} \phi$, is adequate.*

An intersection type interpretation on an applicative structure naturally induces a *logical equivalence*:

Definition 8. Let $\mathcal{A} = (A, \bullet_\mathcal{A}, \Downarrow_\mathcal{A}, [\![\ ]\!]^{t\mathcal{A}})$ be an applicative σ-structure with type interpretation, the *logical equivalence* $\approx^\mathcal{L}_\mathcal{A} \subseteq A \times A$ is defined as follows

$$a \approx^\mathcal{L}_\mathcal{A} b \iff \forall \phi.\ a \in [\![\phi]\!]^{t\mathcal{A}} \Leftrightarrow b \in [\![\phi]\!]^{t\mathcal{A}}.$$

Definition 9. Let $\mathcal{A} = (A, \bullet_\mathcal{A}, \Downarrow_\mathcal{A})$ be an applicative σ-structure. An applicative σ-substructure with convergence of $\mathcal{A}$ is an applicative σ-structure $\mathcal{A}' = (A', \bullet_{\mathcal{A}'}, \Downarrow_{\mathcal{A}'})$, such that $A' \subseteq A$, $\bullet_{\mathcal{A}'} = \bullet_{\mathcal{A}|A' \times A'}$, and $\Downarrow_{\mathcal{A}'} = \Downarrow_{\mathcal{A}|A'}$.

Theorem 10. *Let $\mathcal{A} = (A, \bullet_\mathcal{A}, \Downarrow_\mathcal{A}, [\![\ ]\!]^{t\mathcal{A}})$ be an applicative σ-structure with adequate type interpretation. Let $\mathcal{A}' = (A', \bullet_{\mathcal{A}'}, \Downarrow_{\mathcal{A}'})$ be an applicative σ-substructure of $\mathcal{A}$ such that $\approx^{app}_{\mathcal{A}'} = {\approx^{app}_{\mathcal{A}}}_{|A' \times A'}$. If, for all $\phi \in TB_\sigma$, and for all $a, b \in A$, $a \approx^{app}_\mathcal{A} b \;\Rightarrow\; (a \in [\![\phi]\!]^{t\mathcal{A}} \Leftrightarrow b \in [\![\phi]\!]^{t\mathcal{A}})$, then $\approx^{app}_{\mathcal{A}'} \subseteq \approx^\mathcal{L}_{\mathcal{A}'}$.*

Approximable σ-applicative structures are important structures, because, under the hypothesis of Theorem 10 on the applicative substructure, the applicative equivalence on them coincides with the logical equivalence:

Definition 10. Let $\mathcal{A} = (A, \bullet_\mathcal{A}, \Downarrow_\mathcal{A}, [\![\ ]\!]^{t\mathcal{A}})$ be an applicative σ-structure with type interpretation, and $\mathcal{A}' = (A', \bullet_{\mathcal{A}'}, \Downarrow_{\mathcal{A}'})$ be an applicative σ-substructure. $\mathcal{A}'$ is *approximable* if, for all $a, b_1, \ldots, b_n \in A'$, if $ab_1 \ldots b_n \Downarrow_{\mathcal{A}'}$, then

$$\begin{cases} \exists \phi_1, \ldots, \phi_n \in T_\sigma\ \exists \phi \in T_\sigma^{conv}.\ (a \in [\![\phi_1 \to \ldots \phi_n \to \phi]\!]^{t\mathcal{A}} \wedge & \\ \forall i.\ b_i \in [\![\phi_i]\!]^{t\mathcal{A}}) & \text{if (1)} \\ \exists \phi_1, \ldots, \phi_n \in T_\sigma\ \exists \phi \in T_\sigma^{conv}.\ (a \in [\![\phi_1 \to \ldots \phi_n \to \phi]\!]^{t\mathcal{A}} \wedge & \\ \forall i.\ (b_i \Downarrow_\mathcal{A}\ \wedge b_i \in [\![\phi_i]\!]^{t\mathcal{A}})) & \text{if (2)} \end{cases}$$

where: (1) $\Downarrow_\mathcal{A}$ is not right strict w.r.t. $\bullet_\mathcal{A}$; (2) $\Downarrow_\mathcal{A}$ is right strict w.r.t. $\bullet_\mathcal{A}$.

D^σ with the type interpretation $[\![\ ]\!]^{tD^\sigma}$ of Proposition 9 is approximable.

Theorem 11. *Let $\mathcal{A} = (A, \bullet_\mathcal{A}, \Downarrow_\mathcal{A}, [\![\ ]\!]^{t\mathcal{A}})$ be an applicative σ-structure with adequate type interpretation. Let $\mathcal{A}' = (A', \bullet_{\mathcal{A}'}, \Downarrow_{\mathcal{A}'})$ be an approximable applicative σ-substructure of $\mathcal{A}$:*
i) Then we have $\approx^{app}_{\mathcal{A}'} \supseteq \approx^\mathcal{L}_{\mathcal{A}'}$.
ii) Moreover, if $\mathcal{A}$ and $\mathcal{A}'$ are such that $\approx^{app}_{\mathcal{A}'} = {\approx^{app}_{\mathcal{A}}}_{|A' \times A'}$ and, $\forall \phi \in TB_\sigma, \forall a, b \in A$, $a \approx^{app}_\mathcal{A} b \;\Rightarrow\; (a \in [\![\phi]\!]^{t\mathcal{A}} \Leftrightarrow b \in [\![\phi]\!]^{t\mathcal{A}})$, then $\approx^{app}_{\mathcal{A}'} = \approx^\mathcal{L}_{\mathcal{A}'}$.

Theorem 12. *Let $\mathcal{A} = (A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}}, [\![\ ]\!]^{t\mathcal{A}})$ be an applicative σ-structure with adequate type interpretation. Let $\mathcal{A}' = (A', \bullet_{\mathcal{A}'}, \Downarrow_{\mathcal{A}'})$ be an approximable applicative σ-substructure of $\mathcal{A}$. If $\mathcal{A}$ and $\mathcal{A}'$ are such that $\approx^{app}_{\mathcal{A}'} = \approx^{app}_{\mathcal{A}}\,_{|A' \times A'}$, and for all $\phi \in TB_\sigma$, for all $a, b \in A$, $a \approx^{app}_{\mathcal{A}} b \;\Rightarrow\; (a \in [\![\phi]\!]^{\mathcal{A}} \Leftrightarrow b \in [\![\phi]\!]^{\mathcal{A}})$, then $\approx^{app}_{\mathcal{A}'}$ is a congruence w.r.t. $\bullet_{\mathcal{A}'}$.*

Proof. It is sufficient to show that, for all $k \geq 0$, for all $d_1, \ldots, d_k \in A'$, $abd_1 \ldots d_k \Downarrow_{\mathcal{A}'} \;\Rightarrow\; acd_1 \ldots d_k \Downarrow_{\mathcal{A}'}$.
From $abd_1 \ldots d_k \Downarrow_{\mathcal{A}'}$, by approximability of $\mathcal{A}'$, there exist $\phi_0, \ldots, \phi_k \in T_\sigma$, $\phi \in T^{conv}_\sigma$ s.t. $a \in [\![\phi_0 \to \ldots \phi_k \to \phi]\!]^{t\mathcal{A}} \;\wedge\; b \in [\![\phi_0]\!]^{t\mathcal{A}}$ and $\forall i = 1, \ldots, k.\ b_i \in [\![\phi_i]\!]^{t\mathcal{A}}$.
By Theorem 11, $c \in [\![\phi_0]\!]^{t\mathcal{A}}$, hence $acd_1 \ldots d_k \in [\![\phi]\!]^{t\mathcal{A}}$, and $acd_1 \ldots d_k \Downarrow_{\mathcal{A}'}$. □

The notion of *combinatory σ-algebra* generalizes the standard notion of combinatory λ-algebra (see e.g. [2]).

Definition 11 (Combinatory σ-algebra). A combinatory σ-algebra $\mathcal{A}$ is a structure $(A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}}, [\![\ ]\!]^{\mathcal{A}})$ such that
$(A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}})$ is an applicative σ-structure with convergence;
$[\![\ ]\!]^{\mathcal{A}} : \Lambda(C) \times Env \to A$ is an interpretation function such that, for

$$Env \ni \rho : Var \to \begin{cases} \{a \in A \mid a \Downarrow_{\mathcal{A}}\} & \text{if } \bullet_{\mathcal{A}} \text{ is right strict} \\ A & \text{otherwise} \end{cases}$$

- $[\![x]\!]^{\mathcal{A}}_\rho = \rho(x)$
- $[\![MN]\!]^{\mathcal{A}}_\rho = [\![M]\!]^{\mathcal{A}}_\rho \bullet_{\mathcal{A}} [\![N]\!]^{\mathcal{A}}_\rho$
- $[\![\lambda x.M]\!]^{\mathcal{A}}_\rho \bullet_{\mathcal{A}} a = [\![M]\!]^{\mathcal{A}}_{\rho[a/x]}$, for all $a \in \{a \in A \mid a \Downarrow_{\mathcal{A}}\}$, if $\bullet_{\mathcal{A}}$ is right strict, for all $a \in A$, otherwise;
- $\forall M \in \Lambda^0(C).\ (M \Downarrow_\sigma \;\Rightarrow\; [\![M]\!]^{\mathcal{A}} \Downarrow_{\mathcal{A}})$.

A combinatory σ-algebra $\mathcal{A}$ is *adequate* w.r.t. D^σ, if
$\forall M \in \Lambda^0(C).\ ([\![M]\!]^{\mathcal{A}} \Downarrow_{\mathcal{A}} \Rightarrow\; [\![M]\!]^{D^\sigma} \Downarrow_{D^\sigma})$.

The kind of combinatory σ-algebras we are interested in are those which arise from applicative structures with adequate type interpretations.

Theorem 13 (Soundness). *Let $\mathcal{A}$ be a combinatory σ-algebra with adequate $\mathcal{T}_\sigma$-interpretation. If, for all axioms ax in SB_σ, and $\forall M \in \Lambda(C)$, Γ, ρ,*
$\Gamma \vdash_\sigma M : \phi \;\; ax \;\Longrightarrow\; \forall \rho\ (\forall x.\rho(x) \in [\![\Gamma(x)]\!]^{t\mathcal{A}} \;\Rightarrow\; [\![M]\!]^{\mathcal{A}}_\rho \in [\![\phi]\!]^{t\mathcal{A}})$.
Then, for all $\phi \in T_\sigma$, and for all $M \in \Lambda(C)$, Γ,

$$\Gamma \vdash_\sigma M : \phi \;\Longrightarrow\; \forall \rho\ (\forall x.\rho(x) \in [\![\Gamma(x)]\!]^{t\mathcal{A}} \;\Rightarrow\; [\![M]\!]^{\mathcal{A}}_\rho \in [\![\phi]\!]^{t\mathcal{A}})\ .$$

Proof. The proof is by induction on the length of the derivation $\Gamma \vdash_\sigma M : \phi$. We consider the case $\bullet_{\mathcal{A}}$ not right strict (the other case is similar).
Base Case (axioms in SB_σ): immediate.
Inductive Step: If the last rule in the derivation of $\Gamma \vdash_\sigma M : \phi$ is $\wedge I$, the thesis follows immediately from the induction hypothesis. If the last rule is $\leq$, then the

thesis follows from the induction hypothesis and Definition 7. If the last rule is var, then the thesis follows from the definition of the interpretation of variables in a combinatory σ-algebra.
If the last rule is $\to I$, i.e. $\dfrac{\Gamma[\phi/x] \vdash_\sigma M : \psi}{\Gamma \vdash_\sigma \lambda x.M : \phi \to \psi}$, then, by induction hypothesis, $\forall \rho\ (\forall y.\rho(y) \in [\![\Gamma[\phi/x](y)]\!]^{t\mathcal{A}} \Rightarrow [\![M]\!]^{\mathcal{A}}_{\rho} \in [\![\psi]\!]^{t\mathcal{A}})$. We have to show that $\forall \rho\ (\forall y.\rho(y) \in [\![\Gamma(y)]\!]^{t\mathcal{A}} \Rightarrow [\![\lambda x.M]\!]^{\mathcal{A}}_{\rho} \in [\![\phi \to \psi]\!]^{t\mathcal{A}})$, i.e., if ρ is such that $\forall y.\ \rho(y) \in [\![\Gamma(y)]\!]^{t\mathcal{A}}$, then $\forall a \in [\![\phi]\!]^{t\mathcal{A}}.\ [\![\lambda x.M]\!]^{\mathcal{A}}_{\rho} \bullet_{\mathcal{A}} a \in [\![\psi]\!]^{t\mathcal{A}}$. But, from the definition of term interpretation in a combinatory σ-algebra, we have $[\![\lambda x.M]\!]^{\mathcal{A}}_{\rho} \bullet_{\mathcal{A}} a = [\![M]\!]^{\mathcal{A}}_{\rho[a/x]}$, and moreover $\forall y.\ \rho[a/x](y) \in [\![\Gamma[\phi/x](y)]\!]^{t\mathcal{A}}$. Hence, using the induction hypothesis, we get the thesis.
Finally, if the last rule is $\to E$, i.e. $\dfrac{\Gamma \vdash_\sigma M : \phi \to \psi \quad \Gamma \vdash_\sigma N : \phi}{\Gamma \vdash_\sigma MN : \psi}$, then, by induction hypothesis, $\forall \rho\ (\forall y.\rho(y) \in [\![\Gamma(y)]\!]^{t\mathcal{A}} \Rightarrow ([\![M]\!]^{\mathcal{A}}_{\rho} \in [\![\phi \to \psi]\!]^{t\mathcal{A}} \wedge [\![N]\!]^{\mathcal{A}}_{\rho} \in [\![\phi]\!]^{t\mathcal{A}}))$. Hence $[\![M]\!]^{\mathcal{A}}_{\rho} \bullet_{\mathcal{A}} [\![N]\!]^{\mathcal{A}}_{\rho} \in [\![\psi]\!]^{t\mathcal{A}}$. □

The model D^σ is a combinatory σ-algebra with adequate $\mathcal{T}_\sigma$-interpretation. Moreover, D^σ is sound and complete w.r.t. the type assignment system $\mathcal{T}_\sigma$:

Theorem 14 (Soundness and Completeness). *For all $M \in \Lambda(C), \Gamma$,*

$$\Gamma \vdash_\sigma M : \phi \iff \forall \rho\ (\forall x.\ \rho(x) \in [\![\Gamma(x)]\!]^{tD^\sigma} \Rightarrow [\![M]\!]^{D^\sigma}_{\rho} \in [\![\phi]\!]^{tD^\sigma})\ .$$

Proof. The only non trivial fact is that D^σ with term interpretation $[\![\ ]\!]^{D^\sigma}$ is a combinatory σ-algebra. This is shown following the argument for D^h in [3]. □

Remark. The type assignment system $\mathcal{T}_\sigma$ is sound and complete w.r.t. the class of combinatory σ-algebras with adequate type interpretation, which satisfy the hypotheses of Theorem 13. This is the generalization of the *Completeness* of [3].

Definition 12. Let $\mathcal{A}$ be a combinatory σ-algebra. D^σ is *computationally adequate w.r.t.* $\mathcal{A}$ if, for all $M \in \Lambda^0(C)$, $[\![M]\!]^{\mathcal{A}} \Downarrow_{\mathcal{A}} \iff [\![M]\!]^{D^\sigma} \Downarrow_{D^\sigma}$.

Item ii) of the following theorem generalizes Proposition 7.2.4 of [1].

Theorem 15. *Let $\mathcal{A} = (A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}}, [\![\ ]\!]^{\mathcal{A}})$ be a combinatory σ-algebra with an adequate $\mathcal{T}_\sigma$-interpretation $[\![\ ]\!]^{t\mathcal{A}}$ such that (1) the Soundness Theorem holds for $\mathcal{A}$, and (2) $\mathcal{A}$ is adequate w.r.t. the filter model D^σ. Then*
i) D^σ is computationally adequate w.r.t. $\mathcal{A}$;
ii) let $\mathcal{A}^0 = (A^0, \bullet_{\mathcal{A}^0}, \Downarrow_{\mathcal{A}^0})$ be the applicative σ-substructure of $\mathcal{A}$, where $A^0 \subseteq A$ denotes the interpretation domain of closed λ-terms. Then $\mathcal{A}^0$ is approximable.

Proof. i) Use Theorem 14.
ii) We carry out the proof in the case $\bullet_{\mathcal{A}}$ is not right strict (the other case is similar). We show that, for all $M, N_1, \ldots, N_k \in \Lambda^0(C)$,
$[\![MN_1 \ldots N_k]\!]^{\mathcal{A}} \Downarrow_{\mathcal{A}^0} \implies \exists \phi_1, \ldots, \phi_k \in T_\sigma, \exists \phi \in T^{conv}_\sigma.$
$([\![M]\!]^{\mathcal{A}} \in [\![\phi_1 \to \ldots \phi_k \to \phi]\!]^{t\mathcal{A}} \wedge \forall i.\ [\![N_i]\!]^{\mathcal{A}} \in [\![\phi_i]\!]^{t\mathcal{A}})$. From $[\![MN_1 \ldots N_k]\!]^{\mathcal{A}} \Downarrow_{\mathcal{A}^0}$,

since $\mathcal{A}$ is adequate w.r.t. D^σ, it follows that $[\![MN_1 \dots N_k]\!]^{D^\sigma} \Downarrow_{D^\sigma}$. By approximability of D^σ, $\exists \phi_1, \dots, \phi_k \in T_\sigma, \exists \phi \in T_\sigma^{conv}$ such that
$[\![M]\!]^{D^\sigma} \in [\![\phi_1 \to \dots \phi_k \to \phi]\!]^{tD^\sigma} \ \wedge \ \forall i = 1, \dots, k.\ [\![N_i]\!]^{D^\sigma} \in [\![\phi_i]\!]^{tD^\sigma}$.
From the completeness of D^σ, $\exists \phi_1, \dots, \phi_k \in T_\sigma, \exists \phi \in T_\sigma^{conv}$ such that
$\vdash_\sigma M : \phi \to \dots \phi_k \to \phi \ \wedge \ \forall i = 1, \dots, k.\ \vdash_\sigma N_i : \phi_i$.
By soundness of $\mathcal{A}$, $[\![M]\!]^{\mathcal{A}} \in [\![\phi_1 \to \dots \phi_k \to \phi]\!]^{t\mathcal{A}} \ \wedge \ \forall i = 1, \dots, k.\ [\![N_i]\!]^{\mathcal{A}} \in [\![\phi_i]\!]^{t\mathcal{A}}$. □

Finally, after having presented all these technical results, we are in the position of explaining how to use these results in order to show that the applicative equivalence $\approx_\sigma^{app}$ is a congruence. For the reader's convenience, we outline the:

General pattern of the proof that $\approx_\sigma^{app}$ is a congruence:
1. Let $\to_{\beta_\sigma}$ be a notion of β-reduction which is correct w.r.t. $\approx_\sigma$. Naturally endow $\Lambda(C)/=_{\beta_\sigma}$ (or $\Lambda^0(C)/=_{\beta_\sigma}$) with a structure of combinatory σ-algebra with convergence, i.e. $\mathcal{A}_\sigma = (\Lambda(C)/=_{\beta_\sigma}, \bullet_{=_{\beta_\sigma}}, \Downarrow_\sigma, [\![\ ]\!]^{\mathcal{A}_\sigma})$, where

- by abuse of notation, $\Downarrow_\sigma$ denotes convergence of $=_{\beta_\sigma}$-classes;
- $[\![\ ]\!]^{\mathcal{A}_\sigma} : \Lambda(C) \times Env \to \Lambda(C)/=_{\beta_\sigma}$;
- $Env \ni \rho : Var \to \Lambda(C)/=_{\beta_\sigma}$, if $\bullet_{=_{\beta_\sigma}}$ is not right strict w.r.t. $\Downarrow_\sigma$
  $Env \ni \rho : Var \to Val_\sigma/=_{\beta_\sigma}$, otherwise;
- $[\![M]\!]_\rho^{\mathcal{A}_\sigma} = [\rho(M)]_{=_{\beta_\sigma}}$, where
  $\rho(M) = M[\rho(x_1)/x_1, \dots, \rho(x_n)/x_n]$, with $FV(M) = \{x_1, \dots, x_n\}$.

2. Endow $\mathcal{A}_\sigma$ with an adequate interpretation of the type theory $\mathcal{T}_\sigma$.
3. Use Theorem 15ii) for showing that the applicative σ-substructure of $\mathcal{A}_\sigma$, $\mathcal{A}_\sigma^0 = (\Lambda^0(C)/=_{\beta_\sigma}, \bullet_{=_{\beta_\sigma}}, \Downarrow_\sigma)$, is approximable. (Use Theorem 15i to derive the computational adequacy of D^σ).
4. Use Theorem 12 to show that the σ-structure $\mathcal{A}_\sigma^0$ is such that the applicative equivalence $\approx_{\mathcal{A}_\sigma^0}^{app}$ is a congruence w.r.t. $\bullet_{=_{\beta_\sigma}}$.
5. Use correctness of $\to_{\beta_\sigma}$ reduction w.r.t. $\approx_\sigma$ to deduce that $\approx_\sigma^{app}$ is a congruence w.r.t. λ-term application.

Domain Logic Method: Case $\boldsymbol{\sigma = v}$. In this section we apply the general pattern of the "domain logic method" to the case of the $\to_v$-strategy. Our presentation somewhat simplifies that of [1]. We do not require Λ^0 to be endowed with a structure of lazy call-by-value model, but simply with a v-combinatory algebra structure. Since $\Downarrow_v$ is axiomatized on closed λ-terms, we consider the following syntactical combinatory v-algebra $\mathcal{A}_v^0$ on closed λ-terms:

Definition 13. Let $\mathcal{A}_v^0 = (\Lambda^0/=_{\beta_v}, \bullet_{=_{\beta_v}}, \Downarrow_v, [\![\ ]\!]^{\mathcal{A}_v^0})$ be the combinatory v-algebra defined as follows:
• $[\![\ ]\!]^{\mathcal{A}_v^0} : \Lambda \times Env \to \Lambda^0/=_{\beta_v}$, where $Env \ni \rho : Var \to Val_v/=_{\beta_v}$
• $[\![M]\!]_\rho^{\mathcal{A}_v^0} = [\rho(M)]_{=_{\beta_v}}$, where
$\rho(M) = M[\rho(x_1)/x_1, \dots, \rho(x_n)/x_n]$, with $FV(M) = \{x_1, \dots, x_n\}$.

Now we define a notion of type interpretation on applicative v-structures with convergence, which is proved to be adequate on the whole class of applicative v-structures with convergence. Thus the type assignment system $\mathcal{T}_v$ is sound and complete w.r.t. a large class of structures (see the Remark after Theorem 14).

Definition 14. Let $\mathcal{A} = (A, \bullet_{\mathcal{A}}, \Downarrow_{\mathcal{A}})$ be an applicative v-structure with convergence. We define the type interpretation as follows:
$[\![\nu]\!]^{t\mathcal{A}} = \{a \in A \mid a \Downarrow_{\mathcal{A}}\}$
$[\![\phi \wedge \psi]\!]^{t\mathcal{A}} = [\![\phi]\!]^{t\mathcal{A}} \cap [\![\psi]\!]^{t\mathcal{A}}$
$[\![\phi \to \psi]\!]^{t\mathcal{A}} = \{a \in A \mid a \Downarrow_{\mathcal{A}} \ \wedge \forall c \in [\![\phi]\!]^{t\mathcal{A}}. \ (c \Downarrow_{\mathcal{A}} \Rightarrow \ a \bullet_{\mathcal{A}} c \in [\![\psi]\!]^{t\mathcal{A}})\}$.

Theorem 16. *The notion of type interpretation of Definition 14 is adequate on the whole class of applicative v-structures with convergence. Moreover, it coincides with the canonical type interpretation on the model D^v.*

Applying Theorem 15, we get:

Theorem 17. *i) D^v is computationally adequate.*
ii) $\mathcal{A}_v^0$ is approximable.

Theorem 18. *$\approx_v^{app}$ is a congruence w.r.t. application.*

Proof. From Theorem 12 it follows that $\approx_{\mathcal{A}_v^0}^{app}$ is a congruence w.r.t. application. Hence, using the correctness of $\to_{\beta_v}$ w.r.t. $\approx_v^{app}$, we get the thesis. $\square$

The Domain Logic Method for $\boldsymbol{\sigma = h, n, p, o}$. The application of this method to $\sigma = h, n, p, o$ differs from the one above in two main respects:
• We have to take into account also open terms for $\sigma = h, n, p$, and define accordingly the combinatory σ-algebra $\mathcal{A}_\sigma$, with adequate type interpretation, on the whole $\Lambda / =_{\beta_\sigma}$. Moreover, for $\sigma = h, n$, in order to prove the adequacy of the type interpretation, we need to assume that D^σ is computationally adequate.
• Because of the "non-initial" nature of the model D^σ, for $\sigma = h, n, p, o$, and hence the presence of equations over base types in the corresponding intersection type description, the natural type interpretation on $\mathcal{A}_\sigma$ does not coincide with the canonical type interpretation on D^σ. Hence we have to consider two different type interpretations, the first adequate on $\mathcal{A}_\sigma$ and such that the Soundness Theorem holds, the second adequate on D^σ and realizing both soundness and completeness (Theorem 14).

4.2 Logical Relations Method

This is a semantical method, which uses a computationally adequate CPO-model, obtained as inverse limit. The core of this method consists in defining a suitable relation $\lhd_\sigma \subseteq D^\sigma \times \Lambda^0(C)$. This definition usually requires a deep analysis of the inverse limit model. Then, one proceeds to show

1. $\lhd_\sigma$ is a congruence w.r.t. application;
2. $M \leq^{app}_\sigma N \iff [\![M]\!]^{D^\sigma} \lhd_\sigma N$.

This method extends that originally introduced by Pitts in [16, 17, 18], so that it can be applied to all the strategies mentioned in the Introduction. A related technique appears in [5] for $\to_v$. The technique in [5] (which applies also to $\to_o$), requires to know already that the model is computationally adequate, and also the λ-definability of the projections of the inverse limit. This technique is based on the construction of a relation $\sqsubseteq_\sigma \subseteq \Lambda^0(C) \times \Lambda^0(C)$, which plays the role of the relation $\lhd_\sigma$. The relation $\sqsubseteq_\sigma$ is built inductively in terms of relations $\sqsubseteq^n_\sigma \subseteq (\Lambda^0(C))_n \times (\Lambda^0(C))_n$, where $(\Lambda^0(C))_n$ denotes the set of n-th projections of closed λ-terms. An interesting by-product of this technique is a *purely syntactical* mixed induction-coinduction principle for establishing $\approx^{app}_\sigma$ ([8]).

The technique presented in [16, 17, 18] does not require the λ-definability of projections, but it requires the initiality of D^σ, i.e. the validity of the following "minimal invariance property" à la Plotkin: D^σ is an invariant object of a suitable functor F_σ on the category $CPO_\perp$ via the isomorphism $i_\sigma : D^\sigma \to F_\sigma(D^\sigma)$, and, moreover, id_{D^σ} is the least fixed point of the continuous function $\delta_\sigma : [D^\sigma \to D^\sigma] \to [D^\sigma \to D^\sigma]$ defined as $\delta_\sigma(e) = i^{-1}_\sigma \circ F_\sigma(e) \circ i_\sigma$. Categorically, this amounts to say that D^σ is the initial algebra for a suitable functor on the algebraically compact category $CPO_\perp$ ([6]). This last property guarantees the derivability of an induction-coinduction principle characterizing the relation $\lhd_\sigma \subseteq D^\sigma \times \Lambda^0(C)$. This mixed principle is the essential ingredient for proving that $M \leq^{app}_\sigma N \iff [\![M]\!]^{D^\sigma} \lhd_\sigma N$. Unlike in [5], where $\sqsubseteq_\sigma$ is explicitly characterized inductively in terms of relations $\sqsubseteq^n_\sigma$, here the relation $\lhd_\sigma$ is implicitly defined as the *unique* fixed point of an appropriate mixed variance operator. Hence the relation $\lhd_\sigma$ is a congruence w.r.t. application by definition. The computational adequacy of D^σ can be obtained as a by-product of the method in [16, 17, 18], while it is required for the case of [5]. Because of the requirement that D^σ is initial, Pitts' technique can be applied directly only to $\to_l$ and $\to_v$.

The logical relations method which we present here can be viewed as a generalization and a strengthening of both the two techniques above. We follow the pattern of [16, 17, 18], aiming at the derivation of a suitable induction-coinduction principle for the relation $\lhd_\sigma$, but without assuming the strong minimal invariance property. This is achieved by giving an explicit inductive characterization of $\lhd_\sigma$ in terms of relations $\lhd^n_\sigma \subseteq D^\sigma_n \times \Lambda^0(C)$, in the line of [5]. This technique can be applied successfully to all the strategies $\to_\sigma$ mentioned in the Introduction. It is not always the case, however, that the computational adequacy of the model need not to be assumed at the outset (e.g. for $\sigma = n$).

General pattern of the proof that $\approx^{app}_\sigma$ is a congruence using the logical relations method:
1. Define relations $\unlhd^n_\sigma \subseteq D^\sigma_n \times \Lambda^0(C)$, $n \in \mathbf{N}$, where C contains constants representing the points ϕ of D^σ_0. A constant $c_\phi \in C$ representing ϕ is necessary when no term in Λ^0 is typable with ϕ. The relations $\unlhd^n_\sigma$ are defined inductively, in terms of $\Downarrow_\sigma$. Let $\unlhd_\sigma = \bigcup_{n\in\mathbf{N}} \unlhd^n_\sigma$. Moreover, let $\lhd^n_\sigma = \{(d, P) \in$

$D^\sigma \times \Lambda^0(C) \mid d_n \trianglelefteq^n_\sigma P\}$, and $\lhd_\sigma = \bigcap_{n\in\mathbf{N}} \lhd^n_\sigma$.

Definition 15. A relation $R \subseteq D^\sigma \times \Lambda^0(C)$ is called *limit-closed*, if, whenever for all $n \in \mathbf{N}$ $(d_n, P) \in R$, then also $(d, P) \in R$.

Given a relation $R \subseteq D^\sigma \times \Lambda^0(C)$, $\overline{R}$ denotes the least limit-closed relation including R. The following lemma should then be provable:

Lemma 19. *$\lhd_\sigma$ is the least limit-closed relation including $\trianglelefteq_\sigma$.*

2. Define a monotone "mixed variance" operator $T_\sigma : \mathcal{R}_\sigma \to \mathcal{R}_\sigma$, where $\mathcal{R}_\sigma$ is a complete lattice of pairs of relations such that
i) $\mathcal{R}_\sigma \subseteq \{((R^-, R^+), (\supseteq, \subseteq)) \mid R^-, R^+ \subseteq \bigcup_{n\in\mathbf{N}}(D^\sigma_n \times \Lambda^0(C)) \wedge R^-, R^+ \supseteq \trianglelefteq^0_\sigma\}$,
ii) for all $(R^-, R^+) \in \mathcal{R}_\sigma$,
$(\trianglelefteq^0_\sigma, \trianglelefteq^0_\sigma) \subseteq T_\sigma(R^-, R^+)$
$(d, P) \in \pi_1(T_\sigma(R^-, R^+)) \implies \forall (d', P') \in R^+.\ (dd', PP') \in R^-$
$(d, P) \in \pi_2(T_\sigma(R^-, R^+)) \implies \forall (d', P') \in R^-.\ (dd', PP') \in R^+$.
3. Prove that $(\trianglelefteq_\sigma, \trianglelefteq_\sigma)$ is the least fixed point of T_σ on $\mathcal{R}_\sigma$, i.e.
$\trianglelefteq_\sigma = \bigcup_{R^- \subseteq \pi_1(T_\sigma(R^-,R^+))\quad R^+ \supseteq \pi_2(T_\sigma(R^-,R^+))} R^-$ and
$\trianglelefteq_\sigma = \bigcap_{R^- \subseteq \pi_1(T_\sigma(R^-,R^+))\quad R^+ \supseteq \pi_2(T_\sigma(R^-,R^+))} R^+$.
The proof will be carried out by computing the least fixed point of T_σ explicitly.
4. Since $\lhd_\sigma = \overline{\trianglelefteq_\sigma}$, deduce the following induction-coinduction principle:

$$\frac{R^- \subseteq \pi_1(T_\sigma(R^-, R^+)) \quad R^+ \supseteq \pi_2(T_\sigma(R^-, R^+))}{\overline{R^-} \subseteq \lhd_\sigma \subseteq \overline{R^+}}\ .$$

5. Prove that $\lhd_\sigma$ is a congruence w.r.t. application.
6. Finally prove that $[\![P']\!]^{D^\sigma} \lhd_\sigma P \iff P' \leq^{app}_\sigma P \quad (*)$.
The following two results are instrumental:
i) let $FV(M) \subseteq \{x_1, \ldots, x_n\}$; then

$$\forall i = 1, \ldots, n.\ d_i \lhd_\sigma P_i \implies [\![M]\!]^{D^\sigma}_{\rho[d_i/x_i]} \lhd_\sigma M[P_i/x_i],$$

ii)
$$d \sqsubseteq_{D^\sigma} d' \lhd_\sigma P' \leq^{app}_\sigma P \implies d \lhd_\sigma P.$$

Item ii) is proved using the induction-coinduction principle for $\lhd_\sigma$. To show implication ($\Rightarrow$) in $(*)$ reason by coinduction, using the coinductive characterization of $\leq^{app}_\sigma$. The converse follows immediately from items i) and ii).
7. Deduce that $\leq^{app}_\sigma$ is a congruence w.r.t. application. Moreover, from $\approx^{app}_\sigma = \approx_\sigma$, using items i) and ii) above, deduce that D^σ is computationally adequate.

Logical Relations Method: Case $\sigma = h$. Since no closed λ-term is typable with 1, in order to define the relations $\trianglelefteq^n_h$, we need to consider the set of closed λ-terms extended with a constant 1 denoting the type 1. The relation $\trianglelefteq^n_h$ will be defined on $D^h_n \times \Lambda^0(\{1\})$. The constant 1 plays the role of a generic free variable; hence the extended evaluation $\Downarrow_h \subseteq \Lambda(\{1\}) \times Val_h(\{1\})$ is axiomatized by adding the following rule to those of Section 2: $\dfrac{}{1M_1 \ldots M_m \Downarrow_h 1M'_1 \ldots M'_m}\ m \geq 0$

Definition 16. Define inductively the relations $\trianglelefteq_h^n \subseteq D_n^h \times \Lambda^0(\{1\})$ as:

$$\trianglelefteq_h^0 = \{(\bot, P) \mid P \in \Lambda^0(\{1\})\} \cup \{(1,P) \mid \forall k \geq 0.\ \forall N_1 \dots N_k \in \Lambda^0(\{1\}).\ PN_1 \dots N_k \Downarrow_h\}$$

$$\trianglelefteq_h^{n+1} = \{(d,P) \in D_{n+1}^h \times \Lambda^0(\{1\}) \mid d =_{D^h} \bot \ \vee (d \neq_{D^h} \bot, 1 \ \wedge \ \forall (d',P') \in \trianglelefteq_h^n.\ (dd', PP') \in \trianglelefteq_h^n) \vee (d =_{D^h} 1 \ \wedge \ \forall k \geq 0.\ \forall N_1, \dots, N_k \in \Lambda^0(\{1\}).\ PN_1 \dots N_k \Downarrow_h)\}.$$

Lemma 20. *i)* $\forall d \in D_n^h,\ P \in \Lambda^0(\{1\}),\ d \trianglelefteq_h^n P \implies \forall m.\ d_m \trianglelefteq_h^m P.$
ii) $\triangleleft_h = \underline{\trianglelefteq}_h$.

Proof. i) If $m \geq n$, then the thesis is immediate, by the definition of D_n^h. We are left to show that, for all $n \in \mathbf{N}$, $d_{n+1} \trianglelefteq_h^{n+1} P \implies d_n \trianglelefteq_h^n P$. This is proved by induction, using the well known facts: for all $d, d' \in D^h$, $(d_{n+1}d')_n = (d_{n+1}d'_n)_n = d_{n+1}d'_n = (dd'_n)_n$.
If $n = 0$, the thesis follows immediately by the definitions of $\trianglelefteq_h^0$ and $\trianglelefteq_h^1$.
For $n+1 > 1$, suppose $\bot, 1 \neq d_{n+1} \trianglelefteq_h^{n+1} P$. Take $(d'_{n-1}, P') \in \trianglelefteq_h^{n-1}$. Using the equality above, $(d'_{n-1}, P') \in \trianglelefteq_h^n$, and hence $(d_{n+1}d'_{n-1}, PP') \in \trianglelefteq_h^n$. By induction hypothesis, $(d_n d'_{n-1}, PP') = ((d_{n+1}d'_{n-1})_{n-1}, PP') \in \trianglelefteq_h^{n-1}$. Hence $d_n \trianglelefteq_h^n P$.
ii) The thesis follows from item i). □

A crucial point in applying the logical relations method is to make sure that the least fixed point of the natural operator T_σ associated to the domain D^σ is $(\trianglelefteq_\sigma, \trianglelefteq_\sigma)$. To this end, we give:

Proposition 21. *i) Let $\mathcal{R}_h$ be defined as follows:*

$$\mathcal{R}_h = \{((R^-, R^+), (\supseteq, \subseteq)) \mid R^-, R^+ \subseteq \bigcup_{n \in \mathbf{N}} (D_n^h \times \Lambda^0(\{1\})) \ \wedge \trianglelefteq_h^0 \subseteq R^-, R^+ \ \wedge \ A \cap R^- = \emptyset \ \wedge \ A \cap R^+ = \emptyset\}\ ,$$

where $A = \{(d,P) \in \bigcup_{n \in \mathbf{N}} (D_n^h \times \Lambda^0(\{1\})) \mid d =_{D^h} 1 \ \wedge \ \exists k \geq 0.\ \exists N_1, \dots, N_k \in \Lambda^0(\{1\}).\ PN_1 \dots N_k \not\Downarrow_h\}$. Then $\mathcal{R}_h$ is a complete lattice.
ii) The following operator $T_h : \mathcal{R}_h \to \mathcal{R}_h$ is well defined: $T_h(R^-, R^+) =$
$(\{(d,P) \mid d =_{D^h} \bot \ \vee (d \neq_{D^h} \bot, 1 \ \wedge P \Downarrow_h \ \wedge \forall (d',P') \in R^+.\ (dd', PP') \in R^-) \vee (d =_{D^h} 1 \ \wedge \ \forall k \geq 0.\ \forall N_1, \dots, N_k \in \Lambda^0(\{1\}). PN_1 \dots N_k \Downarrow_h)\} \cap \bigcup_{n \in \mathbf{N}} (D_n^h \times \Lambda^0(\{1\})),$
$\{(d,P) \mid d =_{D^h} \bot \ \vee (d \neq_{D^h} \bot, 1 \ \wedge \ P \Downarrow_h \ \wedge \forall (d',P') \in R^-.\ (dd', PP') \in R^+) \ \vee (d =_{D^h} 1 \ \wedge \ \forall k \geq 0.\ \forall N_1, \dots, N_k \in \Lambda^0(\{1\}). PN_1 \dots N_k \Downarrow_h)\} \cap \bigcup_{n \in \mathbf{N}} (D_n^h \times \Lambda^0(\{1\})))$.

Theorem 22. *i) $(\trianglelefteq_h, \trianglelefteq_h)$ is the least fixed point of T_h.*
ii) The following induction-coinduction principle holds:

$$\frac{R^- \subseteq \pi_1(T_h(R^-, R^+)) \quad R^+ \supseteq \pi_2(T_h(R^-, R^+))}{\overline{R^-} \subseteq \triangleleft_h \subseteq \overline{R^+}}\ .$$

Proof. i) Using Lemma 20i), one can prove that $(\trianglelefteq_h, \trianglelefteq_h)$ is a fixed point of T_h. Moreover, one can check that, for all $n > 0$,
$T_h^{n+1}(\bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})) \setminus A, \trianglelefteq_h) = (\lhd_h^n \cap \bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})), \trianglelefteq_h^n)$.
This follows from the fact that, for $n > 0$,
$T(\lhd_h^n \cap \bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})), \lhd_h^n) = (\lhd_h^{n+1} \cap \bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})), \lhd_h^{n+1})$.
Hence the least fixed point of T_h is
$(\bigcap_{n\in\mathbf{N}} T_h^n(\bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\}))\setminus A, \trianglelefteq_h), \bigcup_{n\in\mathbf{N}} T_h^n(\bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})), v)) =$
$(\lhd_h \cap \bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\})), \trianglelefteq_h) = (\trianglelefteq_h, \trianglelefteq_h)$.
ii) The proof follows from Lemma 20ii) and item i). □

Theorem 23. $d \lhd_h P \iff \forall e \lhd_h Q.\ de \lhd_h PQ$.

Proof. ($\Rightarrow$) The proof follows from the definition of application in the inverse limit model D^h: for all $d, e \in D^h$, $de = \bigsqcup_{n\in\mathbf{N}} d_{n+1}e_n$. Suppose that $d \lhd_h P$ and $e \lhd_h Q$. We will prove that, for all n, $(de)_n \lhd_h PQ$. From the hypotheses and from the definition of $\lhd_h$, using the definition of application, we immediately get, for all n, k, $((\bigsqcup_{m\in\mathbf{N}} d_{m+1}e_m)_n)_k \trianglelefteq_h^k PQ$, i.e., from the definition of $\lhd_h$, we have, for all n, $(\bigsqcup_{m\in\mathbf{N}} d_{m+1}e_m)_n \lhd_h PQ$.
($\Leftarrow$) From the Definition of $\lhd_h$. □

Let $\leq_h^{app,1}$ be the natural extension of $\leq_h^{app}$ to $\Lambda^0(\{1\}) \times \Lambda^0(\{1\})$, i.e. put $M \leq_h^{app,1} N$ iff $\forall P_1 \ldots P_n \in \Lambda^0(\{1\}).\ (MP_1 \ldots P_n \Downarrow_h \Rightarrow NP_1 \ldots P_n \Downarrow_h)$.

Lemma 24. *For all* $d, d' \in D^h$, *and for all* $P, P' \in \Lambda^0(\{1\})$,
$$d \sqsubseteq_{D^h} d' \lhd_h P' \leq_h^{app,1} P \implies d \lhd_h P.$$

Proof. Apply the induction-coinduction principle to $R^+ = \trianglelefteq_h$, $R^- = \{(d, P) \mid \exists d' \in D^h\ \exists P' \in \Lambda^0(\{1\}).\ d \sqsubseteq_{D^h} d' \lhd_h P' \leq_h^{app,1} P\} \cap \bigcup_{n\in\mathbf{N}}(D_n^h \times \Lambda^0(\{1\}))$. □

The following crucial theorem can be easily shown by induction on M, using Theorem 23:

Theorem 25. *Let* $M \in \Lambda(\{1\})$, *with* $FV(M) \subseteq \{x_1, \ldots, x_n\}$. *Then*
$$\forall i = 1, \ldots, n.\ d_i \lhd_h P_i \implies [\![M]\!]^{D^h}_{\rho[d_i/x_i]} \lhd_h M[P_i/x_i].$$

Corollary 26. *For all* $P \in \Lambda^0(\{1\})$, $\quad [\![P]\!]^{D^h} \lhd_h P$.

Lemma 27. $[\![P']\!]^{D^h} \lhd_h P \iff P' \leq_h^{app,1} P$.

Proof. The implication ($\Rightarrow$) can be proved by coinduction. In fact it is easy to check, using Corollary 26 and Theorem 23 ($\Rightarrow$), that $\{(P', P) \mid [\![P']\!]^{D^h} \lhd_h P\}$ is a $\Psi_h^{\leq}$-bisimulation. The proof of the implication ($\Leftarrow$) follows from Lemma 24, and from Corollary 26. □

Theorem 28. *i)* $\leq_h^{app}$ *is a congruence w.r.t. application.*
ii) D^h *is computationally adequate.*

Proof. i) From Theorem 23 and Lemma 27 it follows that $\leq_h^{app,1}$ is a congruence w.r.t. application. Then the thesis follows from the fact that $(\leq_h^{app,1})_{|\Lambda^0 \times \Lambda^0} = \leq_h^{app}$, which in turn is a consequence of Corollary 7.
ii) Using Corollary 26 and Lemma 24, one can show that, if $[\![M]\!]^{D^h} \sqsubseteq_h [\![N]\!]^{D^h}$, then $[\![M]\!]^{D^h} \lhd_h N$, hence by Lemma 27 $M \leq_h^{app} N$. Using i) and Theorems 6, and 8, we have $M \leq_h N$. □

5 Final Remarks

In this paper we introduced two general semantical methodologies for deriving a coinductive characterization of the observational equivalence. But other techniques are also possible. We mention the following three:
1. *Plain induction* on *computation steps* of $\rightarrow_\sigma^*$. This, purely syntactical, direct approach easily applies to $\rightarrow_l$ (see [1]). With suitable extensions in order to take care of open terms, it applies also to $\sigma = h, n$. However, this direct approach is rather problematic for call-by-value strategies such as $\rightarrow_v$, $\rightarrow_p$, or non deterministic strategies like $\rightarrow_o$. For a subtle but complex proof by induction on computation for $\rightarrow_v$ see [15].
2. *Coinductive argument* on *congruence candidate relations.* This method was introduced by D.Howe for the lazy call-by-name strategy $\rightarrow_l$, and later generalized to a class of *lazy* strategies by-name and by-value, including $\rightarrow_v$ ([11]). In [14], we generalize and strengthen Howe's method so as to deal with strategies whose evaluation relations cannot be axiomatized only on closed terms.
3. Method based on the *Separability algorithm.* This method works only if a suitable "separability algorithm" (see e.g. [12, 2]) is available. I.e. if there exists an effective method which, given two non $\approx_\sigma$-equivalent terms, M, N, allows to define an applicative context $C[\ ]$ such that either $C[M] \Downarrow_\sigma$ and $C[M] \not\Downarrow_\sigma$, or viceversa. At present, such an algorithm exists only for $\approx_h$, $\approx_n$, and $\approx_p$. Hence the method works only for $\rightarrow_h$, $\rightarrow_n$, and $\rightarrow_p$.

References

1. S.Abramsky, L.Ong, *Full Abstraction in the Lazy Lambda Calculus,* Information and Computation, 105(2):159–267, 1993.
2. H.Barendregt, *The Lambda Calculus, its Syntax and Semantics,* North Holland, Amsterdam, 1984.
3. H.Barendregt, M.Coppo, M.Dezani-Ciancaglini, *A filter lambda model and the completeness of type assignment,* J.Symbolic Logic, 48(4):931–940, 1983.
4. M.Coppo, M.Dezani-Ciancaglini, M.Zacchi, *Type Theories, Normal Forms and D_∞-Lambda-Models,* Information and Computation, 72(2):85–116, 1987.
5. L.Egidi, F.Honsell, S.Ronchi Della Rocca, *Operational, denotational and logical Descriptions: a Case Study,* Fundamenta Informaticae, 16(2):149–169, 1992.
6. P.Freyd, *Algebraically complete categories,* A.Carboni et al. eds, Category Theory '90 Springer LNM, 1488:95–104, Como, 1990.

7. F.Honsell, M.Lenisa, *Some Results on Restricted λ-calculi,* MFCS'93 Conference Proceedings, A.Borzyszkowski et al. eds., Springer LNCS, 711:84–104, 1993.
8. F.Honsell, M.Lenisa, *Final Semantics for untyped λ-calculus,* M.Dezani et al. eds, TLCA'95 Springer LNCS, 902:249–265, Edinburgh, 1995.
9. F.Honsell, M.Lenisa, *A Semantical Analysis of an Effective Perpetual Strategy,* Draft, November 1996.
10. F.Honsell, S.Ronchi Della Rocca, *An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus*, J. of Computer and System Sciences, 45(1):49-75, 1992.
11. D.Howe, *Proving Congruence of Bisimulation in Functional Programming Languages*, Information and Computation, 124(2):103–112, 1996.
12. M.Hyland, *A survey of some useful partial order relations on terms of the lambda-calculus,* C.Böhm ed., Springer LNCS, 37:83–95, 1975.
13. M.Lenisa, *Final Semantics for a Higher Order Concurrent Language,* CAAP'96 Conference Proceedings, H.Kirchner ed., Springer LNCS, 1059:102–118, 1996.
14. M.Lenisa, *The Congruence Candidate Method for Giving Coinductive Characterizations of Observational Equivalences in λ-calculi,* to appear in CAAP'97 Proc..
15. I.Mason, S.Smith, C.Talcott, *From Operational Semantics to Domain Theory,* Information and Computation to appear.
16. A.M.Pitts, *Computational Adequacy via 'Mixed' Inductive Definitions,* MFPS'93, Brookes et al. eds., Springer LNCS, 802:72–82, 1994.
17. A.M.Pitts, *A Note on Logical Relations Between Semantics and Syntax*, to appear in WoLLIC'96 Proc., J. of the Interest Group in Pure and Applied Logics.
18. A.M.Pitts, *Relational Properties of Domains*, Information and Computation, 127:66–90, 1996.
19. G.D.Plotkin, *Call-by-name, Call-by-value and the λ-calculus,* Theoretical Computer Science 1:125-159, 1975.