Must preorder in non-deterministic untyped λ-calculus*

U. de'Liguoro and A. Piperno

Dipartimento di Matematica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma
deliguoro@vaxrma.infn.it
piperno@vaxrma.infn.it

Abstract

This paper studies the interplay between functional application and nondeterministic choice in the context of untyped λ-calculus. We introduce an operational semantics which is based on the idea of *must* preorder, coming from the theory of process algebras. To characterize this relation, we build a model using the classical inverse limit construction, and we prove it fully abstract using a generalization of Böhm trees.

1 Introduction

The study of call-by-name and call-by-value passing mechanisms was tackled, in the framework of untyped λ-calculus, in the pioneering work by Plotkin [18]. More recently, it has been reformulated in a categorical setting by Moggi [16], while lazyness, that is a special case of call-by-name, has been the main issue in the research by Abramsky and Ong [1]. A number of papers has appeared therafter on this topic, which seems to be an area of growing interest.

At the same time, in the field of concurrency theory, the need for a notion of application, and consequently of abstraction, has been recognized (implicitly [15], or explicitly [22]), so that new problems arise, particularly as far as a combination of non-determinism and functional application is concerned.

λ-calculi enriched with non-deterministic operators are not a novelty (see [2, 3]), but studies on a type free calculus are rather rare (see [6, 20, 14]). In Sharma's thesis [20], inspired by the work of Hennessy [10], the problem of finding a correct extension of the notion of λ-theory in the case of a λ-calculus enriched with a choice operator has been connected with the distinction of two value passing mechanisms (i.e. β-rules), which can be

*This work has been partially supported by grants from ESPRIT-BRA 3230.

viewed as restrictions to the choice operator of call-by-name and call-by-value evaluation strategies. The first rule allows unrestricted substitution, in such a way that choices can be multiplied and performed at any moment during evaluation; this is called *run-time-choice* in [10]. The second one forces passed values to be "deterministic", eliminating choices in a term before substituting it for an abstracted variable: this is *call-time-choice* in [10].

Combining methods from classical, type-free λ-calculus [13, 23] with those from the theory of process algebras [9], we investigate on a kind of *must testing* preorder, written $\sqsubseteq_{must}$. Usually the *may* preorder is considered to model parallelism in the functional setting (see [6]), but the must preorder seems to be more deeply connected with correctness problems arising from the value passing mechanism.

However, there is a stronger reason for choosing the idea of must convergency in the λ-calculus: as a matter of fact it turns out to be the right extension of the notion of *solvability*, central in the classical theory, leading to a model and to a theory which nicely includes sensible ones, and precisely which is a conservative extension of $\mathcal{H}^*$, the unique Hilbert-Post complete extension of the λ-theory equating all unsolvable terms (see [4]).

After introducing the reduction relation $\longrightarrow_r$ in section 2, we give a characterization of the operational semantics induced by must preorder, defined in section 3, by means of a model; the model is built in section 4 using a construction based on a powerdomain functor (see [19, 21]) and non deterministic algebras [2, 11]. The main result of the paper is that, for each pair of terms M and N of our calculus,

$$M \sqsubseteq_{must} N \;\Leftrightarrow\; [\![M]\!] \sqsubseteq [\![N]\!]$$

where $[\![\cdot]\!]$ is the interpretation map: i.e. a full abstraction theorem. This is proved in section 5 introducing a kind of trees similar to Böhm trees [4], and defining an order relation over them which turns out to be equivalent to those in the main theorem. Finally in section 6 the theory of the model is studied, and it is shown that it conservatively extends $\mathcal{H}^*$. As a byproduct, we prove the consistency of the full theory λ_r, introduced in [20].

We assume familiarity with the classical theory of the λ-calculus and with the powerdomain construction.

2 The non-deterministic λ-calculus

In this section we shortly review two possible ways of adding a choice operator to the type-free λ-calculus, somehow related to *call-by-name* and *call-by-value* reduction strategies.

Definition 1 *(Syntax)*
The syntax of the non-deterministic λ-calculus is defined as follows:

$$M ::= x \mid \lambda x.M \mid MN \mid M \oplus N.$$

Terms, whose set is denoted by $\Lambda_\oplus$, are considered modulo α-congruence; clearly $\Lambda \subset \Lambda_\oplus$.

Following [10, 20], we distinguish two parameter-passing mechanisms, namely two β-rules:

Definition 2 *(Rules)*

(i) *Run-time choice*

(β_r) $(\lambda x.M)N \to M[N/x]$,
where $(M \oplus N)[L/x] \equiv M[L/x] \oplus N[L/x]$;

(i') *Call-time choice*

(β_c) $(\lambda x.M)N \to M[N/x]$ if $N \in \Lambda$;

(ii)
(μ) $N \to N' \Rightarrow MN \to MN'$,
(ν) $M \to M' \Rightarrow MN \to M'N$,
(ξ) $M \to M' \Rightarrow \lambda x.M \to \lambda x.M'$,
(η) $\lambda x.Mx \to M$ *if* $x \notin FV(M)$,

$(\oplus.1)$ $M \to M' \Rightarrow M \oplus N \to M' \oplus N$,
$(\oplus.2)$ $N \to N' \Rightarrow M \oplus N \to M \oplus N'$,
$(\oplus.3)$ $M \oplus N \to M$,
$(\oplus.4)$ $M \oplus N \to N$.

The rules (β_c) and (β_r) yield two different reduction relations, namely $\longrightarrow_c$ and $\longrightarrow_r$. We are interested in the study of the interplay between the functional application and the choice operator, which is fully formalized by the run-time choice reduction; the interest in the aspect of functionality explains also the rule (η). On the other hand, as we will see in the final section, a calculus based on call-time choice is nothing more than a calculus whose objects are finite sets of classical terms.

We shall concentrate our attention on run-time-choice reduction relation. In the following, $\longrightarrow$ will stand for $\longrightarrow_r$.

Despite the lack of the Church-Rosser property, a standardization theorem can be proved. In general one cannot permute β-reductions with $\oplus$-reductions:

$$\begin{tikzcd}
(\lambda x.xx)(P \oplus Q) \arrow[r, "\beta"'] \arrow[d, "\oplus"'] & (P \oplus Q)(P \oplus Q) \arrow[d, "\oplus"', "*"] \\
(\lambda x.xx)P \quad ?? & PQ
\end{tikzcd}
\qquad
\begin{tikzcd}
((\lambda x.P) \oplus Q)R \arrow[d, "\oplus"'] & ?? \\
(\lambda x.P)R \arrow[r, "\beta"'] & P[R/x]
\end{tikzcd}$$

Figure 1: β and $\oplus$-reductions do not always commute

indeed, figure 1 shows that a β-contraction can multiply $\oplus$-redexes (left), hence the number of possible choices, and that $\oplus$-contractions may create new β-redexes (right).

However, if the choice does not delete a β-redex, whose residual[1] is contracted thereafter, then these reduction steps commute. This is the key to prove the following proposition:

Proposition 1 *Let* $\mathcal{F} \subseteq M$ *be a set of redex occurrences in* M*, with* $\mathcal{F} = \mathcal{F}_\beta \cup \mathcal{F}_\oplus$ *where* $\mathcal{F}_\beta$ *and* $\mathcal{F}_\oplus$ *are respectively the* β*- and* $\oplus$*-redexes in* $\mathcal{F}$*; then every development of* $\mathcal{F}$ *is finite; moreover*

[1] For the notions of *residual* and *development*, as well as for the notation, see [4].

$$M \xrightarrow[dev]{\mathcal{F}} N \ \Rightarrow \exists M'.\ M \xrightarrow[dev]{\mathcal{F}_\beta} M' \xrightarrow[dev]{\mathcal{F}_\oplus} N.$$

The term in which each development ends is in general not unique.

We recall that a reduction is *standard* (STN) iff a residual of a redex whose main operator occurs to the left of the main operator of a contracted redex is never contracted thereafter. Using the previous proposition the following theorem is proved as in the classical case with some minor changes

Theorem 1 *For all* $M, N \in \Lambda_\oplus$. $M \xrightarrow{*} N \Rightarrow M \xrightarrow[STN]{} N$.

The important consequence of this theorem is that any reduction can be transformed into another one consisting of some head reductions followed by internal reductions only.

3 Operational semantics

As it is well known, in the classical λ-calculus a term has no meaning when it doesn't reduce to a head normal form, that is when it is *unsolvable*. In the extension we are interested in, two possible generalizations suggest themselves: the first one says that a term is solvable iff it reduces to a head normal form (we would call it *may convergency*): a study using a similar notion is e.g. [6]. The second one defines a term to be solvable iff it has no infinite head reduction: we call this *must convergency* and write $M{\downarrow}_{must}$ or simply $M{\downarrow}$. Beside any other justification for choosing the latter notion as a research topic, we will show that it naturally leads to a conservative extension of the well known sensible theory $\mathcal{H}^*$.

Inspired by the extensional equivalence of Morris [17] and its analogue by Wadsworth [23] and by the idea of testing given by De Nicola and Hennessy [9] for process algebras, we define the following notions (see also [14]):

Definition 3 *For* $M \in \Lambda_\oplus$ *define*

1. $M{\downarrow}_{must} \Leftrightarrow M$ *has no infinite head reduction,*
2. $M \sqsubseteq_{must} N \Leftrightarrow \forall C[\,].\ C[M]{\downarrow}_{must} \Rightarrow C[N]{\downarrow}_{must}$,
3. $M \simeq_{must} N \Leftrightarrow M \sqsubseteq_{must} N \sqsubseteq_{must} M$.

We write $M{\uparrow}$ *to mean not* $M{\downarrow}$.

It should be noted that $\sqsubseteq_{must}$ is a preorder, so that, taking the quotient under $\simeq_{must}$, we get an order which is a precongruence.

Furthermore, such order is sensitive to the choice structure of a term, as the following example shows.

Example 1 let $M \equiv \lambda x.x(y \oplus z)$ and $N \equiv (\lambda x.xy) \oplus (\lambda x.xz)$, and consider the context $C[\,] \equiv (\lambda yz.[\,])H_0H_1\Delta$, with $H_0 \equiv \lambda x.x\mathbf{U}^3_3\Delta$ and $H_1 \equiv \lambda xy.yy$, $\Delta \equiv \lambda x.xx$, $\mathbf{U}^3_3 \equiv$

$\lambda x_1 x_2 x_3 . x_3$; a simple computation shows that $C[M]\!\uparrow$ while $C[N]\!\downarrow$; hence $N \not\sqsubseteq_{must} M$: it will be proved that $M \sqsubseteq_{must} N$.

Finally, we note that, if $M \xrightarrow{*} N$ then $M \sqsubseteq_{must} N$, i.e. the order increases under reduction.

4 Denotational semantics

In order to characterize the relation $\sqsubseteq_{must}$, we build a model, which is however of interest on its own. The structure we are looking for is an applicative structure with an extra operator modelling $\oplus$.

Definition 4 *A* semilinear applicative structure *is a triple* $\langle X, \cdot, + \rangle$ *such that*

1. $\langle X, \cdot \rangle$ *is an applicative structure,*
2. $+ : X^2 \to X$ *is an idempotent, commutative and associative operation,*
3. $\forall x, y, z \in X.\ (x + y) \cdot z = (x \cdot z) + (y \cdot z)$.

We call it *semilinear* since we do not allow in general the application to be right distributive too, that is to be linear. This is due to the fact that the application will be used to model functions whose argument are "sets", that is sums, and it is not true in general that the value of these functions is the set of their values on the "elements" of the argument.

To get a fully abstract interpretation of $\Lambda_\oplus$-terms with respect to $\sqsubseteq_{must}$, we have to consider the following issues:

- to build a semilinear applicative structure as an ordered set where the $+$ operation satisfies the Smyth's axiom:
$$x + y \sqsubseteq x;$$
- the structure has not to be linear, i.e.
$$x \cdot (y + z) \neq (x \cdot y) + (x \cdot z);$$
- the stucture has to be extensional, i.e.:
$$(\forall z.\ x \cdot z = y \cdot z) \Rightarrow x = y,$$
since $\simeq_{must}$ validates the equation:
$$\lambda x. Mx = M \quad \text{if } x \notin FV(M).$$

To fulfill the first requirement, we work in the category of Smyth Non-Deterministic Algebras, **SNDA** (see below), which is cartesian closed (see [2]). The second point, however, rules out any solution of the domain equation $D = D \to_{lin} D$, which yields a linear structure. Finally, the extensionality requirement prewents the use of Moggi's construction (see [16]), involving an equation of the form $D = \wp D \to_{cont} \wp D$, to be solved in the category of **CPO**.

We will solve, instead, the equation $D = \wp D \to_{lin} D$ in the category **SNDA**, using the fact that $\wp D \to_{lin} D$ is isomorphic, as a **CPO**, with $D \to_{cont} D$: this satisfies extentionality as well as semilinearity and non-linearity.

4.1 The domain D_*

Our aim is to solve the domain equation

$$D = \wp D \rightarrow_{lin} D$$

where $\wp$ is the Smyth powerdomain functor, in the category of **SNDA**. The objects of this category are of the form $\langle D, \sqsubseteq, + \rangle$, where $\langle D, \sqsubseteq \rangle$ is a domain, and $+$ is a binary continuous operation which is idempotent, commutative, associative and such that $x+y \sqsubseteq x$ holds for each $x, y \in D$: we call them *Smyth algebras*. Morphisms are continuous functions preserving $+$; we write $\rightarrow_{lin}$ in contrast to $\rightarrow_{cont}$, which is referred to continuous funtions.

We recall that, given $\langle D_1, \sqsubseteq_1, +_1 \rangle$ and $\langle D_2, \sqsubseteq_2, +_2 \rangle$, the cartesian product $D_1 \times D_2$ is defined as the product of the domains D_1 and D_2, with the opration $+$ defined:

$$\langle x, y \rangle + \langle x', y' \rangle = \langle x +_1 x', y +_2 y' \rangle,$$

while the exponentiation $D_1 \rightarrow_{lin} D_2$ is formed taking the set of linear continuous functions from D_1 to D_2, pointwise ordered, and defining:

$$\forall x \in D_1.\, (f+g)(x) = f(x) +_2 g(x).$$

SNDA can be proved to be cartesian closed.

Finally, we recall that, given any domain D, $\wp D$ is the free Smyth algebra generated by D.

Definition 5 *Take D_0 as any non trivial Smyth algebra (e.g. $\wp\mathbf{2}$), and $D_{n+1} = [\wp D_n \rightarrow_{lin} D_n]$; then inductively define $\varphi_n : D_n \rightarrow_{lin} D_{n+1}$ and $\psi_n : D_{n+1} \rightarrow_{lin} D_n$ as follows:*

1. $\varphi_0(x) = \lambda y.x$, $\psi_0(y) = y(\perp)$,
2. $\varphi_{n+1}(x) = \varphi_n \circ x \circ \wp\psi_n$, $\psi_{n+1}(y) = \psi_n \circ y \circ \wp\varphi_n$.

The $\mathcal{O}$-functoriality of $\wp$ (see e.g. [11]), that is

$$\forall f, g.\, f \sqsubseteq g \Rightarrow \wp f \sqsubseteq \wp g,$$

where $\sqsubseteq$ is the pointwise ordering, ensures that $\langle \varphi_n, \psi_n \rangle$ is an embedding-projection pair, for each n, so that we can take

$$D_* = \lim_{\leftarrow} (D_n, \psi_n).$$

As usual with the inverse limit construction, each D_n embeds into D_*, say by the $\langle \Phi_{n,*}, \Phi_{*,n} \rangle$ embedding-projection pair; setting $x_n = \Phi_{*,n}(x)$ for $x \in D_*$ and $a_n = \wp\Phi_{*,n}(a)$ for $a \in \wp D_*$, we can state

Proposition 2 *The map $F : D_* \rightarrow [\wp D_* \rightarrow_{lin} D_*]$ defined by*

$$F(x) = \lambda a \in \wp D_*.\bigsqcup_n x_{n+1}(a_n)$$

and the map $G : [\wp D_* \rightarrow_{lin} D_*] \rightarrow D_*$ defined by

$$G(f) = \bigsqcup_n (\lambda a \in \wp D_n.(f(\wp\Phi_{n,*}(a)))_n)$$

are continuous, linear and mutually inverse.

4.2 A semilinear applicative structure

By proposition 2, the solution of our domain equation is a Smyth algebra, hence it has an operation $+$ which can interpret our $\oplus$. We now introduce the application operation.

Proposition 3 *For any Smyth algebras D and E,*

$$\wp D \rightarrow_{lin} E \simeq D \rightarrow_{cont} E$$

in the category of **CPO**.

Proof. The isomorphism is given by $\_ \circ \{\!|\cdot|\!\}$ and by $(\cdot)^\dagger$, where, for any $f : D \rightarrow_{cont} E$, $f^\dagger$ is the unique linear continuous function such that $f = f^\dagger \circ \{\!|\cdot|\!\}$.

$\square$

Now, setting $\tilde{F} = (\_ \circ \{\!|\cdot|\!\}) \circ F$ and $\tilde{G} = G \circ (\cdot)^\dagger$, we define the application:

$$x \cdot y = \tilde{F}(x)(y) = F(x)\{\!|y|\!\}.$$

We conclude this section listing some basic properties of the structure D_*.

Lemma 1 *For any $x, y, z \in D_*$ and $a \in \wp D_*$,*

1. $(x + y) \cdot z = (x \cdot z) + (y \cdot z)$,
2. $x = \bigsqcup_n x_n,\ a = \bigsqcup_n a_n$,
3. $x_{n+1} \cdot y_n = x_{n+1}(\{\!|y_n|\!\})$,
4. $x_{n+1} \cdot y = x_{n+1} \cdot y_n = (x \cdot y_n)_n$,
5. $x_0 \cdot y = x_0 = (x \cdot \bot)_0$.

Proof. We prove *1*:

$$\begin{array}{rcll}
(x+y)\cdot z & = & \tilde{F}(x+y)(z) & \\
 & = & F(x+y)(\{\!|z|\!\}) & \\
 & = & (F(x)+F(y))(\{\!|z|\!\}) & \text{by linearity of } F \\
 & = & F(x)(\{\!|z|\!\}) + F(y)(\{\!|z|\!\}) & \\
 & = & (x\cdot z) + (y \cdot z). &
\end{array}$$

The other points are almost the same as in the D_∞ model of the λ-calculus; in *3*, we observe that, for any f, $\wp f \circ \{\!|\cdot|\!\} = \{\!|\cdot|\!\} \circ f$; it follows that $\{\!|y|\!\}_n = \{\!|y_n|\!\}$, so that *3* follows from the definition of $\cdot$ and standard properties of the inverse limit constructions.

$\square$

4.3 A fully abstract interpretation of $\Lambda_\oplus$

We present a *model*, which actually does not directly interpret the relation $\longrightarrow_r$, but the equivalence relation induced by $\sqsubseteq_{must}$.

Definition 6 *A* syntactical model *is a semilinear applicative structure $\langle X, \cdot, + \rangle$, equipped with a map $[\![\cdot]\!] : \Lambda_\oplus \to (Env \to X)$, such that the triple $\langle X, \cdot, [\![\cdot]\!] \rangle$ satisfies the clauses of the classical definition of* syntactical λ-model *of* [12], *and furthermore*

$$[\![M \oplus N]\!]_\rho = [\![M]\!]_\rho + [\![N]\!]_\rho,$$

where $\rho \in Env = Var \to X$.

Remark 1 The equation

$$[\![\lambda x.M \oplus N]\!]_\rho = [\![\lambda x.M]\!]_\rho + [\![\lambda x.N]\!]_\rho$$

is easily proved to be valid if the applicative structure is extensional.

Definition 7 *Given the structure $\langle D_*, \cdot, + \rangle$ and $\rho \in Env = Var \to D_*$, we define the map $[\![\cdot]\!] : \Lambda_\oplus \to (Env \to D_*)$ as follows:*

1. $[\![x]\!]_\rho = \rho(x)$,
2. $[\![MN]\!]_\rho = [\![M]\!]_\rho \cdot [\![N]\!]_\rho$,
3. $[\![\lambda x.M]\!]_\rho = \tilde{G}(\lambda d \in D_*.[\![M]\!]_{\rho[d/x]})$,
4. $[\![M \oplus N]\!]_\rho = [\![M]\!]_\rho + [\![N]\!]_\rho$.

This is a good definition, since in *3* the continuity of application, abstraction and + ensures that the function $\lambda d \in D_*.[\![M]\!]_{\rho[d/x]}$ is continuous as well. Now it can be verified that:

Proposition 4 *The quadruple $\langle D_*, \cdot, +, [\![\cdot]\!] \rangle$ is a syntactical model, furthermore it is extensional.*

We can now state the main theorem of this paper:

Theorem 2 (Full Abstraction Theorem)
For all M, $N \in \Lambda_\oplus$

$$M \sqsubseteq_{must} N \;\Leftrightarrow\; \forall\rho.\, [\![M]\!]_\rho \sqsubseteq [\![N]\!]_\rho.$$

The next section is devoted to the proof of this theorem.

5 Trees for proving full abstraction

In order to represent the functional behaviour of each term, we introduce a kind of unfolding trees, generalizing the notion of Böhm trees of the classical λ-calculus, and a suitable notion of approximation, to be considered as a cut of the tree.

The difficulty here is that, since the Church-Rosser property doesn't hold, we cannot consider our trees as a representation of the directed set of "approximated" reducts of a term: instead, we have to take into account all possible reductions, without making the choices, but simply representing their structure somehow in the tree.

5.1 Non-deterministic Böhm trees and their ordering

As a first step, we introduce a notation to represent approximations.

Definition 8 *We define, by mutual induction, two sets $\mathcal{S}_0$ and $\mathcal{S}_1$:*

1. $\Omega \in \mathcal{S}_0$, *where Ω is a new constant representing divergency,*
2. $M \in \mathcal{S}_0 - \{\Omega\} \Rightarrow \lambda x.M \in \mathcal{S}_0$,
3. $M_1, ..., M_n \in \mathcal{S}_0 \Rightarrow \{M_1, ..., M_n\} \in \mathcal{S}_1 \quad$ *for* $n > 0$,
4. $\mathcal{M}_1, ..., \mathcal{M}_m \in \mathcal{S}_1, x \in Var \Rightarrow x\mathcal{M}_1 ... \mathcal{M}_m \in \mathcal{S}_0 \quad$ *for* $m \geq 0$.

For every approximant, there is a term in $\Lambda_\oplus\Omega$ (i.e., $\Lambda_\oplus$ extended with the constant Ω) which corresponds to it in a natural way; we define, by mutual induction, $\vartheta_0 : \mathcal{S}_0 \to \Lambda_\oplus\Omega$ and $\vartheta_1 : \mathcal{S}_1 \to \Lambda_\oplus\Omega$:

$$\begin{array}{rcl} \vartheta_0(\Omega) = \Omega, & & \vartheta_0(\lambda x.M) = \lambda x.\vartheta_0(M), \\ \vartheta_0(x\mathcal{M}_1 \ldots \mathcal{M}_m) & = & x\vartheta_1(\mathcal{M}_1) \ldots \vartheta_1(\mathcal{M}_m); \\ \vartheta_1(\{M_1, \ldots, M_n\}) & = & \vartheta_0(M_1) \oplus \ldots \oplus \vartheta_0(M_n). \end{array}$$

To simplify the notation, we will identify $\mathcal{M} \in \mathcal{S}_1$ with $\vartheta_1(\mathcal{M}) \in \Lambda_\oplus\Omega$.

Definition 8 allows us to introduce the notion of *Non-deterministic Böhm Tree* (NBT).

Definition 9 *Let $\mathcal{M} \in \mathcal{S}_1$; rather informally, we define:* $\mathrm{NBT}(\mathcal{M}) = \mathrm{NBT}_1(\mathcal{M})$, *where* $\mathrm{NBT}_0(\Omega) = \Omega$ *and:*

$$\mathrm{NBT}_0(\lambda\vec{x}.\xi\mathcal{M}_1 ... \mathcal{M}_m) = \begin{array}{c} \lambda\vec{x}.\xi \\ \diagup \quad \diagdown \\ \mathrm{NBT}_1(\mathcal{M}_1) \ \cdots \ \mathrm{NBT}_1(\mathcal{M}_m) \end{array} \qquad \mathrm{NBT}_1(\{M_1, ..., M_n\}) = \begin{array}{c} \oplus \\ \diagup \quad \diagdown \\ \mathrm{NBT}_0(M_1) \ \cdots \ \mathrm{NBT}_0(M_n) \end{array}$$

Note that the order of sons of a node labelled by $\lambda\vec{x}.\xi$ is relevant, as well as multiple occurrences of the same subtree; this is not the case for sons of nodes labelled by $\oplus$.

NBT's may be seen as infinite $\mathcal{S}_1$-terms, that is as elements of the completion of $\mathcal{S}_1$ under the order induced on $\mathcal{S}_1$ by the relation freely generated by $\Omega \preceq M$, for all M, on $\mathcal{S}_0$. More precisely, they are the limits of those directed subsets of $\mathcal{S}_1$ generated by terms in $\Lambda_\oplus$ with the following family of maps:

Definition 10 *For each natural number k, we define a map ω^k: $\Lambda_\oplus \to \mathcal{S}_1$ by:*

1. $\omega^0(M) = \{\Omega\}$,
2. (a) $\omega^{k+1}(M) = \{\Omega\}$, *if M diverges; otherwise:*
   (b) $\omega^{k+1}(M) = \{\lambda\vec{x}.\xi\, \omega^k(M_1)...\omega^k(M_m) \mid \lambda\vec{x}.\xi M_1...M_m$ *is a principal*[2] *head normal form of* $M\}$.

Furthermore, we denote

$$M^{[k]} = \vartheta_1 \circ \omega^k(M).$$

Remark 2 We note that $M^{[k]}$ is always a β-Ω-normal form (see [4]), where each $\oplus$-redex cannot create new β-Ω-redexes. We denote by $\mathrm{N}^{\Omega}_{\oplus}$ the set of such terms.

Example 2 Let $M \equiv \lambda x.x(yx) \oplus \lambda x.xz \ \in \ \Lambda_\oplus$. Then we have:

$$\begin{array}{rcl}
\omega^1(M) & = & \{\lambda x.x\, \omega^0(y\Omega) \ , \ \lambda x.x\, \omega^0(z)\} = \{\lambda x.x\{\Omega\}\}, \\
\omega^2(M) & = & \{\lambda x.x\, \omega^1(yx) \ , \ \lambda x.x\, \omega^1(z)\} = \{\lambda x.x\{y\, \omega^0(x)\} \ , \ \lambda x.x\{z\}\}, \\
 & = & \{\lambda x.x\{y\{\Omega\}\} \ , \ \lambda x.x\{z\}\}, \\
M^{[1]} & = & \lambda x.x\Omega, \\
M^{[2]} & = & \lambda x.x(y\Omega) \oplus \lambda x.xz.
\end{array}$$

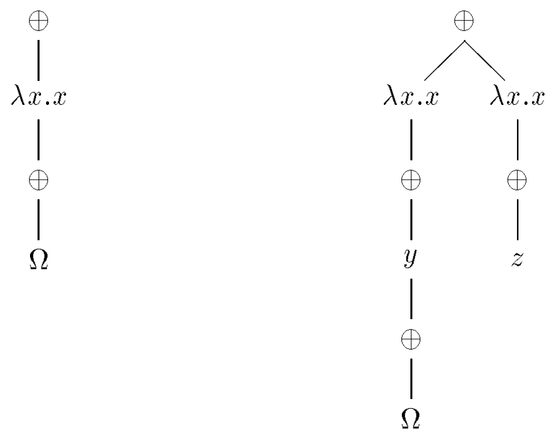

Figure 2: Non-deterministic Böhm trees

The trees $\mathrm{NBT}(\omega^1(M))$ and $\mathrm{NBT}(\omega^2(M))$ are shown in Figure 2.

To compare two terms, that is their trees, simple inclusion doesn't suffice even in the classical λ-calculus: what we need is a generalization of the relation ${}^{\eta}\sqsubseteq^{\eta}$ in [4], or, equivalently, of $<^s_k$ in [13]: this will be achieved in several steps.

We first recall the notion of *equivalence* ($\sim$) for head normal forms [5] (called *similarity* in [13]): given two classical hnf's

$$M \equiv \lambda x_1 \dots x_n.\xi M_1 \dots M_m \quad \text{and} \quad N \equiv \lambda x_1 \dots x_{n'}.\zeta N_1 \dots N_{m'}$$

[2] i.e. it is reachable by head reductions only.

(where m and m' are called *degrees* of M and N) they are equivalent iff $\xi \equiv \zeta$ and $n - m = n' - m'$.

Since the relation $\sim$ is preserved by η-reduction and expansion, we note that, if $M \sim N$, then we can η-expand M and N in such a way that they have the same degree.

The relation $\sim$ plays an important role in comparing elements of $\Lambda_{\oplus}$, as shown by the following example.

Example 3 Let $M \equiv \lambda x_1x_2x_3.x_1x_3(x_2x_3) \oplus \lambda x_1x_2.x_1x_2x_2 \oplus \lambda x_1x_2x_3.x_1x_3x_2$ and $N \equiv \lambda x_1x_2.x_1x_1 \oplus \lambda x_1x_2x_3.x_1x_2$.

We have:

$$\begin{array}{rcl}
\omega^2(M) & = & \{\lambda x_1x_2x_3.x_1\{x_3\}\{x_2\{\Omega\}\} \ , \ \lambda x_1x_2.x_1\{x_2\}\{x_2\} \ , \\
& & \lambda x_1x_2x_3.x_1\{x_3\}\{x_2\}\}; \\
\omega^2(N) & = & \{\lambda x_1x_2.x_1\{x_1\} \ , \ \lambda x_1x_2x_3.x_1\{x_2\}\}; \\
M^{[2]} & = & \lambda x_1x_2x_3.x_1x_3(x_2\Omega) \oplus \lambda x_1x_2.x_1x_2x_2 \oplus \lambda x_1x_2x_3.x_1x_3x_2; \\
N^{[2]} & = & N.
\end{array}$$

It comes out that there exists a $\sim$-equivalence class of $\omega^2(M) \cup \omega^2(N)$ which does not contain any element of $\omega^2(M)$. In this case, we can immediately find a context such that $C[M]$ converges while $C[N]$ does not. Indeed, take $C[\,] \equiv [\,](\lambda a_1a_2a_3.a_1)x_2x_3x_4\Omega$, tehn

$$\begin{array}{rcl}
\omega^2(C[M^{[2]}]) & = & \{x_4\{x_5\}, x_3\{x_4\}\{x_5\}\}, \\
\omega^2(C[N^{[2]}]) & = & \{\Omega\}.
\end{array}$$

In the general case, however, while comparing two terms, it is necessary to analyze their internal structure.

Definition 11 *Given* $\mathcal{M}$, $\mathcal{N} \in \mathcal{S}_1$, *define* $\mathsf{Pair}(\mathcal{M}, \mathcal{N})$ *as the least set such that, if* $\mathcal{M}_{/\sim} = \{\mathcal{E}\} \cup \mathcal{M}'$ *and* $\mathcal{N}_{/\sim} = \{\mathcal{F}\} \cup \mathcal{N}'$, *where* $\mathcal{E} = \{M^1, \ldots, M^m\}$ *and* $\mathcal{F} = \{N^1, \ldots, N^n\}$ *and* $(\mathcal{E} \cup \mathcal{F})_{/\sim}$ *is a singleton, assuming*

$$\begin{array}{ll}
M^i \equiv \lambda\vec{y}.x\mathcal{M}^i_1 \ldots \mathcal{M}^i_l, & \text{for } 1 \le i \le n, \\
N^j \equiv \lambda\vec{y}.x\mathcal{N}^j_1 \ldots \mathcal{N}^j_l, & \text{for } 1 \le j \le m,
\end{array}$$

then, for each $p \le l$,

$$< \{\mathcal{M}^1_p, \ldots, \mathcal{M}^n_p\}, \{\mathcal{N}^1_p, \ldots, \mathcal{N}^m_p\} > \in \mathsf{Pair}(\mathcal{M}, \mathcal{N}).$$

Given $\mathcal{M}, \mathcal{N} \in \mathcal{S}_1$, $\mathsf{Pair}(\mathcal{M}, \mathcal{N})$ selects the subterms to be compared during the first step of the analysis of the internal structure of $\mathcal{M}$ and $\mathcal{N}$. As in [13], this notion has to be generalized to each level of the tree.

Definition 12 *Given* $\mathcal{M}$, $\mathcal{N} \in \mathcal{S}_1$, *define* $\mathsf{Pair}_k(\mathcal{M}, \mathcal{N})$, *for each natural number* k, *as follows:*

1. $\mathsf{Pair}_1(\mathcal{M}, \mathcal{N}) = \mathsf{Pair}(\mathcal{M}, \mathcal{N})$.

2. *Let* $\mathsf{Pair}(\mathcal{M},\mathcal{N}) = \{<\mathcal{U}_1,\mathcal{V}_1>,\ldots,<\mathcal{U}_n,\mathcal{V}_n>\}$ *and* $\mathcal{U}'_i = \bigcup\mathcal{U}_i$, $\mathcal{V}'_i = \bigcup\mathcal{V}_i$, *where* $1 \leq i \leq n$*: then*

$$\mathsf{Pair}_{k+1}(\mathcal{M},\,\mathcal{N}) = \{<\mathcal{A},\mathcal{B}>\mid \exists i \leq n.\ <\mathcal{A},\mathcal{B}>\in\ \mathsf{Pair}_k(\mathcal{U}'_i,\mathcal{V}'_i)\}.$$

Remark 3 Note that in *2*, for $1 \leq i \leq n$, $\mathcal{U}_i$ is a finite non empty set of objects in $\mathcal{S}_1$, hence a family of finite non empty sets of objects in $\mathcal{S}_0$; it follows that its union $\mathcal{U}'_i$ is again an element of $\mathcal{S}_1$.

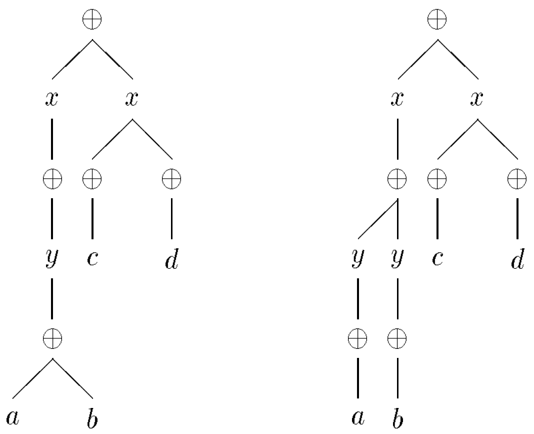

Figure 3: Respectively the trees of $\omega^3(M)$ and $\omega^3(N)$

Example 4 Let $M \equiv x(y(a\oplus b))\oplus xcd$ and $N \equiv x(ya\oplus yb)\oplus xcd$ (see figure 3); we have

$$\begin{array}{rcl} \omega^3(M) & = & \{x\{y\{a,b\}\}\ ,\ x\{c\}\{d\}\} \\ \omega^3(N) & = & \{x\{y\{a\},y\{b\}\}\ ,\ x\{c\}\{d\}\}. \end{array}$$

From this we compute:

$$\begin{array}{rcl} \mathsf{Pair}_1(\omega^3(M),\omega^3(N)) & = \{ & <\{\{y\{a,b\}\}\},\{\{y\{a\},y\{b\}\}\}>, \\ & & <\{\{c\}\},\{\{c\}\}>\ ,\ <\{\{d\}\},\{\{d\}\}>\quad\}; \\ \mathsf{Pair}_2(\omega^3(M),\omega^3(N)) & = & \mathsf{Pair}_1(\{y\{a,b\}\},\{y\{a\},y\{b\}\})\cup \\ & & \mathsf{Pair}_1(\{c\},\{c\})\cup\mathsf{Pair}_1(\{d\},\{d\}) \\ & = \{ & <\{\{a,b\}\},\{\{a\},\{b\}\}>\quad\}. \end{array}$$

We are now ready to introduce the ordering relation $\leq$ over trees.

Definition 13 *Given* $\mathcal{M},\,\mathcal{N}\in\mathcal{S}_1$ *we define, for each* k*, a relation* $\leq_k$ *by:*

$$\begin{array}{rcl} \mathcal{M}\leq_1\mathcal{N} & \Leftrightarrow & \mathcal{M}=\{\Omega\}\ \vee\ \forall N\in\mathcal{N}\ \exists M\in\mathcal{M}.\ M\sim N, \\ \mathcal{M}\leq_{k+1}\mathcal{N} & \Leftrightarrow & \mathcal{M}\leq_k\mathcal{N}\wedge \\ & & \forall<\mathcal{U},\mathcal{V}>\in\mathsf{Pair}_k(\mathcal{M},\mathcal{N}).\mathcal{U}\sqsubseteq^\sharp\mathcal{V}, \end{array}$$

where

$$\mathcal{U} \sqsubseteq^\sharp \mathcal{V} \Leftrightarrow \forall \mathcal{N}_j \in \mathcal{V}\, \exists \mathcal{M}_i \in \mathcal{U}.\, \mathcal{M}_i \leq_1 \mathcal{N}_j.$$

From this we can define

$$\mathcal{M} \leq \mathcal{N} \Leftrightarrow \forall k.\;\; \mathcal{M} \leq_k \mathcal{N}.$$

Finally, for any M*,* $N \in \Lambda_\oplus$*,*

$$\begin{aligned} M \leq_k N \quad &\Leftrightarrow \quad \omega^k(M) \leq_k \omega^k(N), \\ M \leq N \quad &\Leftrightarrow \quad \forall k.\;\; M \leq_k N. \end{aligned}$$

Let us note that the relation $\sqsubseteq^\sharp$ is the Smyth preorder induced by $\leq_1$.

5.2 The semi-separability theorem

The semi-separability problem has been studied, for finite sets of λ-terms, in [7]. However, these techniques are not sufficient in cases where the relation $\sqsubseteq^\sharp$ is essentially involved, as it will be clear from the proof of the following lemma:

Lemma 2 *For* $M, N \in \Lambda_\oplus$*,*

$$M \not\leq_2 N \;\Rightarrow\; \exists C[\,].\, C[M]\Downarrow \;\; \wedge\; C[N]\Uparrow\,.$$

Proof. (Sketch) By cases. If $M \not\leq_1 N$, then either $\omega^1(N) = \mathcal{N} = \{\Omega\}$ and $\omega^1(M) = \mathcal{M} \neq \{\Omega\}$, so that there is nothing to prove, or there exists $N \in \mathcal{N}$ such that, for all $M \in \mathcal{M}$, $M \not\sim N$: then the problem is solved using the semi-separability techniques described in [7].

We exhibit an example of the case where $M \leq_1 N$ but $M \not\leq_2 N$. Take $M \equiv xy \oplus xz$ and $N \equiv x(y \oplus z)$; now $\omega^2(M) = \{x\{y\}, x\{z\}\}$, while $\omega^2(N) = \{x\{y, z\}\}$; computing $\mathsf{Pair}_1(\omega^2(M), \omega^2(N))$, we get $\{< \{\{y\}, \{z\}\}, \{\{y, z\}\} >\}$ and it can be verified that $\{\{y\}, \{z\}\} \not\sqsubseteq^\sharp \{\{y, z\}\}$. If $\mathbf{1} \equiv \lambda uv.uv$ and $\mathbf{2} \equiv \lambda uv.u(uv)$ are two Church numerals, we take $C_0[\,] \equiv (\lambda xyz.[\,])(\lambda w.aww)\mathbf{1}\,\mathbf{2}$. Simple calculations give us

$$\omega^2(C_0[M]) = \{a\{\mathbf{1}\}\{\mathbf{1}\}, a\{\mathbf{2}\}\{\mathbf{2}\}\}$$

and

$$\omega^2(C_0[N]) = \{a\{\mathbf{1},\, \mathbf{2}\}\{\mathbf{1},\, \mathbf{2}\}\}.$$

Now, there exists a combinator $\mathbf{H}$ such that, if L is a sum of Church numerals and $\mathbf{n}$ a numeral, then $\mathbf{1} \in \omega^2(\mathbf{H}L\mathbf{n})$ only if $\mathbf{n}$ occurs in the sum L, it is $\{\mathbf{0}\}$ otherwise[3]. Taking

$$C_1[\,] \equiv (\lambda a.[\,])(\lambda uv.\mathbf{P}(\mathbf{H}u\mathbf{1})(\mathbf{H}v\mathbf{2})),$$

where $\mathbf{P}$ λ-defines multiplication, we have

$$\omega^2(C_1[C_0[M]]) = \{\mathbf{0}\} \;\text{ and }\; \omega^2(C_1[C_0[N]]) = \{\mathbf{0},\, \mathbf{1}\}.$$

[3]Basically $\mathbf{H}$ is a test for equality for Church numerals, with the property that the first argument is not multiplied.

Now, taking $C_2[\,] \equiv [\,](\mathbf{K}\Omega)\mathbf{I}$, we have

$$\omega^2(C_2[C_1[C_0[M]]]) = \{\mathbf{I}\}$$

while

$$\omega^2(C_2[C_1[C_0[N]]]) = \{\Omega\}.$$

$\square$

The following lemma rephrases, in our context, the Böhm-out lemma of [4].

Lemma 3 *For $M, N \in \Lambda_\oplus$ and $k \geq 2$,*

$$M \not\leq_k N \;\Rightarrow\; \exists C[\,].\, C[M] \not\leq_2 C[N].$$

We are now ready to prove the semi-separability theorem for our calculus.

Theorem 3 *For any M, $N \in \Lambda_\oplus$,*

$$M \sqsubseteq_{must} N \;\Rightarrow\; M \leq N.$$

Proof. By contraposition, we prove (see definition 3)

$$\exists k.\, M \not\leq_k N \;\Rightarrow\; \exists C[\,].\, C[M]{\Downarrow} \wedge C[N]{\Uparrow}\,.$$

Indeed,

$$\begin{array}{rcll} M \not\leq N & \Rightarrow & \exists k.\, M \not\leq_k N & \\ & \Rightarrow & \exists C[\,].\, C[M] \not\leq_2 C[N] & \text{by lemma 2} \\ & \Rightarrow & \exists C[\,], C'[\,].\, C'[C[M]]{\Downarrow} \;\wedge\; C'[C[N]]{\Uparrow} & \text{by lemma 3.} \end{array}$$

$\square$

5.3 Proving the full abstraction theorem

Finally, we can prove the last results leading us to the main theorem.

Theorem 4 *For all $M, N \in \Lambda_\oplus$,*

$$\forall \rho \in Env.\, [\![M]\!]_\rho \sqsubseteq [\![N]\!]_\rho \;\Rightarrow M \sqsubseteq_{must} N.$$

Proof. (Sketch) Following a pattern similar to that in [13], one shows that, if $L \in \mathrm{N}_\oplus^\Omega$ and $L \leq M$, then $[\![L]\!] \sqsubseteq [\![M]\!]$, and that $[\![M]\!] = \bigsqcup_k [\![M^{[k]}]\!]$; using these facts we have

$$\begin{array}{rcl} M^{[1]} = \Omega & \Leftrightarrow & \forall k.\, M^{[k]} = \Omega \\ & \Leftrightarrow & \forall k.\, [\![M^{[k]}]\!] = \bot \\ & \Leftrightarrow & [\![M]\!] = \bigsqcup_k [\![M^{[k]}]\!] = \bot \end{array}$$

since $[\![\Omega]\!] = \bot$; hence

$$\begin{array}{rcl} M \not\sqsubseteq_{must} N & \Rightarrow & \exists C[\,].\, C[M]{\Downarrow} \wedge C[N]{\Uparrow} \\ & \Rightarrow & \exists C[\,].\, \omega^1(C[M]) \neq \{\Omega\} = \omega^1(C[N]) \\ & \Rightarrow & \exists C[\,].\, [\![C[M]]\!] \neq \bot = [\![C[N]]\!] \\ & \Rightarrow & [\![M]\!] \not\sqsubseteq [\![N]\!], \end{array}$$

being the context operation the composition of abstraction, application and $\uplus$, that is a monotonic function. $\square$

Corollary 1 *For all* $M, N \in \Lambda_{\oplus}$,

1. $M \leq N \;\Rightarrow [\![M]\!] \sqsubseteq [\![N]\!]$,
2. $M \sqsubseteq_{must} N \;\Leftrightarrow\; M \leq N \;\Leftrightarrow\; [\![M]\!] \sqsubseteq [\![N]\!]$.

Proof. Using the proof of the theorem and the theorem itself, we have

$$\begin{array}{rcl} M \leq N & \Rightarrow & \forall k.\, M^{[k]} \leq M \leq N \\ & \Rightarrow & \forall k.\, [\![M^{[k]}]\!] \sqsubseteq [\![N]\!] \\ & \Rightarrow & [\![M]\!] \sqsubseteq [\![N]\!]. \end{array}$$

This establishes *1*; now *2* follows from *1* and theorems 3 and 4. □

6 Non-Deterministic Theories

The convertibility relations generated by $\longrightarrow_r$ $\longrightarrow_c$ are trivially inconsistent;indeed, in both cases:

$$\forall M, N \in \Lambda_{\oplus}.\;\; M \longleftarrow M \oplus N \longrightarrow N.$$

However, we have just seen that the equational theory

$$\mathcal{T} = \{M = N \mid M \simeq_{must} N\}$$

has a non trivial model, hence it is consistent. In [20] two extensions of the concept of λ-theory have been proposed: λ_r and λ_c.

Definition 14 *The theory* λ_r *is the equational theory over* $\Lambda_{\oplus}$ *whose axioms and rules are as follows*

(β_r) $(\lambda x.M)N = M[N/x]$;

(ρ) $M = M$;

(σ) $M = N \Rightarrow N = M$;

(τ) $M = N, N = L \Rightarrow M = L$;

(μ) $M = N \Rightarrow LM = LN$;

(ν) $M = N \Rightarrow ML = NL$;

(ξ) $M = N \Rightarrow \lambda x.M = \lambda x.N$;

(ζ_1) $M \oplus M = M$;

(ζ_2) $M \oplus N = N \oplus M$;

(ζ_3) $(M \oplus N) \oplus L = M \oplus (N \oplus L)$;

(ϵ) $M = N \Rightarrow M \oplus L = N \oplus L$;

(δ) $(M \oplus N)L = ML \oplus NL$;

The theory λ_c is obtained substituting axiom (β_r) by:

$$(\beta_c) \qquad (\lambda x.M)N = M[N/x] \quad \text{if } N \in \Lambda,$$

and adding the axioms

(θ) $L(M \oplus N) = LM \oplus LN$;

(γ) $\lambda x.M \oplus N = \lambda x.M \oplus \lambda x.N$;

About the theory λ_c we claimed that it is nothing more than a calculus of finite sets of classical terms; more precisely, associating to each term the set of its deterministic behaviours modulo the relation $\longrightarrow_c$, we state the following theorem, whose proof is in [8]:

Theorem 5 *Define* $\mathsf{det}(M) = \{L \in \Lambda \mid M \stackrel{*}{\longrightarrow}_c L\}^+$, *where* $(\cdot)^+$ *means the closure under usual β-conversion, and put $M =_c N$ iff $\mathsf{det}(M) = \mathsf{det}(N)$; then, for any $M, N \in \Lambda_\oplus$*

$$\lambda_c \vdash M = N \Rightarrow M =_c N.$$

It follows that λ_c is a conservative extension of λ.

On the other hand, we can show that the theory $\mathcal{T}$ actually includes the theory λ_r, and it is consistent: hence λ_r is consistent too. The consistency of a subtheory of λ_r was established in [20] by orienting the equations from left to right and showing that the relation obtained in this way is Church-Rosser. However, it was not clear how this theory was related with the notion of *choice*. On the contrary $\mathcal{T}$ is strongly related to the reduction relation $\longrightarrow_r$ and its relation with λ_r is comparable to the relation between $\mathcal{H}^*$ and λ; therefore we can prove the consistency of the *full* theory λ_r from the consistency of $\mathcal{T}$. Furthermore $\mathcal{T}$ is a conservative extension of $\mathcal{H}^*$.

This is achieved by means of the following lemma, whose proof is in [8].

Lemma 4 *Let M, $N \in \Lambda_\oplus$. Then*

$$\exists D[\,] \in \Lambda_\oplus[\,].\ D[M]\!\downarrow \ \wedge\ D[N]\!\uparrow \ \Rightarrow\ \exists C[\,] \in \Lambda[\,].\ C[M]\!\downarrow \ \wedge\ C[N]\!\uparrow.$$

Concluding we have

Theorem 6 *Let $\mathcal{T} = \{M = N \mid M \simeq_{must} N\}$; then:*

1. $\lambda_r \subseteq \mathcal{T}$,
2. *$\mathcal{T}$ is a conservative extension of $\mathcal{H}^*$,*
3. *λ_r and $\mathcal{T}$ are consistent.*

Proof. To prove the first part, we use corollary 1 *2* and the fact that D_* is a model of λ_r. For the second part, let M, $N \in \Lambda$ be such that $\mathcal{T} \not\vdash M = N$; then there is a context $D[\,] \in \Lambda_\oplus[\,]$ such that, say, $D[M]\Downarrow$ and $D[N]\Uparrow$. By lemma 4, there is a context $C[\,] \in \Lambda[\,]$ such that $C[M]\downarrow$, that is $C[M] \in$ **SOL**, the set of solvable terms, and $C[N] \notin$ **SOL**; hence $\mathcal{H}^* \not\vdash M = N$. The third part follows from the first two and the consistency of $\mathcal{H}^*$ (or equivalently of $\mathcal{T}$). □

Acknowledgments

We are greatful to Corrado Böhm, Mariangiola Dezani-Ciancaglini, Gianfranco Mascari and Eugenio Moggi for helpful discussions and suggestions about the topics of this paper.

References

[1] S. Abramsky, C.H.L. Ong, *Full Abstraction in the Lazy Lambda Calculus*, Research Rep., Dept. of Comp., Imperial College 1989.

[2] E.A. Ashcroft, M.C.B. Hennessy, "A mathematical Semantics for a Nondeterministic Typed Lambda Calculus", *TCS* 11, 1980.

[3] E. Astesiano, G. Costa "Distributive Semantics for Nondeterministic Typed λ-calculi", *TCS* 32, 1984.

[4] H.P. Barendregt, *The Lambda-Calculus: Its Syntax and Semantics*, North-Holland, 1984.

[5] C. Böhm, "Alcune proprieta' delle forme β-η-normali nel λ-K-calcolo", Pubblicazioni dell' I.A.C. n. 696, Roma 1968.

[6] G. Boudol, "A Lambda Calculus for Parallel Functions", INRIA Preprint, 1990.

[7] M. Coppo, M. Dezani-Ciancaglini, S. Ronchi della Rocca, "(Semi)-separability of finite sets of terms in Scott's D_∞-models of the λ-calculus", *LNCS* 62, 1978.

[8] U. de'Liguoro, "Non deterministic untyped λ-calculus", Ph.D. Thesis, 1991.

[9] R. De Nicola, M.C.B. Hennessy, "Testing Equivalences for Processes", *TCS* 34, 1983.

[10] M.C.B. Hennessy, "The Semantics of Call-by-value and Call-by-name in a nondeterministic Environment", *SIAM J. Comput.* 9, 1980.

[11] M.C.B. Hennessy, G.D. Plotkin, "Full Abstraction for a Simple Parallel Programming Language", *LNCS* 74, 1979.

[12] J.R. Hindley, G. Longo, "Lambda Calculus Models and Extensionality", *Z. Math. Logik Grundlag. Math.* 26, 1980.

[13] M. Hyland, “ A Syntactic Characterization of the Equality in some Models for the Lambda Calculus”, *J. of the London Math. Soc.* 12, 1976.

[14] R. Jagadeesan, P. Panangaden, “A Domain-theoretic Model for a Higher-order Process Calculus”, *LNCS* 443, 1990.

[15] R. Milner “Functions as Processes”, *LNCS* 443, 1990.

[16] E. Moggi, “Notions of Computation and Monads”, *Inf. Comp.* 93, 1991.

[17] J.H. Morris, *Lambda Calculus Models of Programming Languages*, Dissertation, M.I.T. 1968.

[18] G.D. Plotkin, “Call-by-name, Call-by-value and the λ-calculus”, *TCS* 1, 1975.

[19] G.D. Plotkin, “A Powerdomain Construction”, *SIAM J. of Comp.* 5, 1976.

[20] K. Sharma, *Syntactic Aspects of the Non-deterministic Lambda Calculus*, Master’s thesis, Washington State University, September 1984. Available as internal report CS-84-127 of the Comp. Sci Dept.

[21] M.B. Smyth, “Power Domains”, *J. Comp. Sys. Sci.* 16, 1978.

[22] B. Thomsen, “A Calculus of Higher-Order Communicating Systems”, *ACM* 143, 1989.

[23] C.P. Wadsworth, “ The relation between computational and denotational properties for Scott’s D_∞-models of the lambda-calculus”, *SIAM J. of Comp.* 5, 1976.