Filter Models for Conjunctive-Disjunctive λ-calculi*

Mariangiola Dezani-Ciancaglini [1] *Ugo de'Liguoro* [1] *Adolfo Piperno* [2]

[1] Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, 10149 Torino, Italy
{dezani,deligu}@di.unito.it

[2] Dipartimento di Scienze dell'Informazione
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
piperno@dsi.uniroma1.it

January 24, 1996

Abstract. The distinction between the conjunctive nature of non-determinism as opposed to the disjunctive character of parallelism constitutes the motivation and the starting point of the present work. λ-calculus is extended with both a non-deterministic choice and a parallel operator; a notion of reduction is introduced, extending β-reduction of the classical calculus.
We study type assignment systems for this calculus, together with a denotational semantics which is initially defined constructing a set semimodel via simple types. We enrich the type system with intersection and union types, dually reflecting the disjunctive and conjunctive behaviour of the operators, and we build a filter model. The theory of this model is compared both with a Morris-style operational semantics and with a semantics based on a notion of capabilities.

1 Introduction

A variety of non-deterministic and parallel operators have been added to the λ-calculus by several authors with different aims. One has been the study of non-determinism in the functional setting (see e.g. [7, 14, 2] and more recently [1, 36]), i.e. the study of (computable) multivalued functions. This view is strictly connected with the theory of powerdomains introduced in [38, 43].

These efforts receive new interest in connection with recent research activities aiming at a theory of higher-order communicating processes. So it is natural to ask for a theory in which communication embodies functional application. This has been studied by Thomsen in [44] and by Boudol in [15] explicitly, while it is an implicit theme in current research on Milner's π-calculus [32].

Non-determinism and parallelism (usually represented by an interleaving operator) are fundamental concepts in process algebra theory. Combining them and λ-calculus can enlighten the theory of higher-order process algebras. Indeed an open problem with the former theory is the lack of a good denotational semantics. It is encouraging that a main step toward a definition of

* This work has been partially supported by grants from ESPRIT-BRA 7232 GENTZEN and from CNR-GNASAGA.

what is a model of a higher-order process algebra has been done by Hennessy in [22] by resorting to logical models of type-free lazy λ-calculus. On the other hand higher-order process algebras may be helpful in understanding λ-calculus theories capturing evaluation strategies, like lazy and call-by-value λ-calculi, as shown in [31, 44, 41].

Extensions of the λ-calculus with non-deterministic and/or parallel operators have been also considered in order to gain definability of combinators like Plotkin's parallel-or [37]. These extensions increase the power of the λ-calculus to detect convergence internally (easily done by call-by-value mechanisms) also in those cases in which a term converges as soon as at least one of its subterms does, no matter in which order they are evaluated. This amounts to have the definability of all compact points in a standard model, that is, by Milner's theorem, to have a fully abstract interpretation for the language.

In [16] an analysis of parallel-or in terms of an asynchronous parallel operator ($\|$) and call-by-value abstraction is proposed. Because of this asynchronicity, a term $M\|N$ can be reduced independently on both sides; to make it convergent if and only if M or N are, Boudol defines a term to be convergent if at least one of its possible computations (properly reductions) ends, what is called a *may convergency* notion. In the same paper a fully abstract, denotational semantics is provided for this calculus. This semantics is based on the Stone duality paradigm, implicitly introduced for use in denotational semantics in [42] [13]. This paradigm has been explicitly advocated in [3], where the filter model construction of [13] has been put in its right mathematical setting. A full abstraction theorem is then stated and proved.

The investigation carried out in [16] has been pursued further by the present authors both in a richer setting and in a different perspective in [20]. In that paper we consider the calculus proposed by Ong in [35]. It includes a parallel and a non-deterministic operator, as well as call-by-name and call-by-value abstractions. To gain the expected behaviour, the parallel operator (always denoted by $\|$) is a synchronous operator. The non-deterministic operator (denoted by $+$) is instead an internal choice operator. By synchronicity, a term $M\|N$ is irreducible as soon as M or N is in normal form, and hence there is no need for a may convergency predicate. This choice makes explicit the different meanings of $\|$ and $+$, which are kept distinct by stipulating that a term is convergent if and only if all its reductions eventually stop, that is by using a *must convergency* criterion.

In [20] we construct a denotational model by means of a logical system. This time, however, intersection type discipline does not suffice any more (as in case of [16]). We use also union types, introduced for the classical λ-calculus in [11]. The operators $\|$ and $+$ are respectively interpreted as join and meet over the semantic domain, and they are dually typed by intersection and union. Even in this case a fully abstract semantics is obtained.

[6] defines a powerdomain functor, which has the features of being convex and of preserving algebraic lattices. This allows to give a fully abstract interpretation of a call-by-name and call-by-value lazy λ-calculus enriched with a parallel operator, a non-deterministic operator and an operator mapping a set of terms into its join.

[16], [20], and [6] consider variants of the lazy λ-calculus. The present paper aims instead to study the *full* classical λ-calculus extended with $\|$ and $+$. This essentially amounts to allow reduction under abstraction and evaluation of the argument even before passing it.

Since the original paper [4] by Abramsky and Ong, it has been argued that the lazy λ-calculus is a better model of actual implementations of functional programming languages like Scheme. Indeed these languages do not evaluate the bodies of functions before formal parameters have been replaced by the arguments to which functions are applied. Similarly they do not evaluate the arguments before passing them.

There is, however, a missing point in treating functional languages in a lazy perspective. In that setting we are forced to look at functions in a merely extensional way, that is as black boxes whose different behaviours can be detected just testing them against application to suitable arguments and waiting for the output (but also, possibly, waiting forever). As a matter of fact, the semantics of the lazy λ-calculus has been defined in [4] by introducing the notion of functional bisimulation, which is nothing but a sophisticated version of the extensional idea.

The unfolding semantics (sometimes called algebraic semantics) is a well established theory of recursive languages, originated with Tarsky's fixed point theorem and with Kleene's first recursion theorem. This theory has its λ-calculus counterpart in the notion of Böhm tree, which finely recover topological ideas from the syntactical notions of head normal forms and separability (see [12]). Now it seems that such a theory does not exist in the case of lazy λ-calculus. As a matter of fact, the problem cannot be remedied by resorting to Lévy-Longo trees, since they induce a finer semantics than functional bisimulation (this has been shown in Ong's thesis [34]). This justifies our choice of considering the classical λ-calculus.

In the present paper we give a semantics based on the notion of unfolding for our parallel and non-deterministic extension of classical λ-calculus. This is not achieved by means of trees, but by using the equivalent notion of approximant originated, in the case of λ-calculus, from the works of Lévy [28] and Wadsworth [45].

In the first section of the paper we introduce the syntax of the calculus and two reduction relations. The first one explicitly makes the $+$ into a choice operator, while the second one, instead, simulates the choice by a distribution law. Adapting to the present case the notion of head reduction and head normal form, we prove that both reductions define the same set of solvable terms, so that in the following we study the second reduction relation which is technically easier to handle.

After a short discussion of the contextual theory induced by the set of solvable terms, we define the concept of approximant and the connected notion of capability (reminiscent of the homonymous notion in [35]), formally setting the unfolding semantics that we study.

In the subsequent two sections we introduce a type assignment system in two steps. The first one considers just Curry types, simply adding to the assignment system the rules for typing $M||N$ and $M + N$. As a preliminary result we get Plotkin's set semimodel [39] for our calculus and the equational theory on terms which it induces. We then enrich the type syntax with intersection, union types, and the universal type ω. Types are partially ordered so that they give rise, by the usual filter construction, to a distributive lattice which, as a domain, is an ω-algebraic prime lattice. We refer to [5] for more details and for the description of the domain equation underlying the construction, which involves both lower and upper powerdomain functors, combined with the space of Scott-continuous functions. By adding a subtyping rule and an intersection introduction rule, the type assignment system turns out to be sound and complete with respect to λ-lattices, which are λ-models with a lattice structure.

The last section contains the main results of the paper, namely the approximation theorem and the full abstraction theorem. Roughly speaking, the approximation theorem says that the set of types of any term is the union of all types that can be given to its approximations, hence being the limit of them in the logical semantics. The full abstraction theorem states that the unfolding semantics and the logical semantics are actually the same. Moreover, we get that solvable terms are characterized as those terms which are typeable by a type which is not equivalent to ω.
Some of the results of the present paper were stated in [19], where only the reduction relation here called $\longrightarrow_{pn}$ was considered.

2 Conjunctive and Disjunctive λ-calculus

In this section we give the syntax of our calculus and prove the basic properties of two reduction relations. The general theme is that of distinguishing between non-determinism and parallelism.

It is certainly debatable whether these two notions have to be kept distinct, since in many cases parallelism is explained in terms of non-determinism. This is true in particular when the aim of parallelism is the possibility of handling simultaneously several different computations and of terminating as soon as one of these computations terminates.

But if we implement this device using a choice operator, then we must assume the existence of an oracle which, at each stage, will suggest the right decision. In this way the oracle will prevent any non terminating computation, whenever at least one output of the non-deterministic program exists. This is no more necessary if, instead, we use an operator which does not make choices, but which evaluates in a synchronous way its arguments. I.e. an operator which does one reduction step only when both its arguments are reducible, and which stops otherwise.

On the contrary the choice operator comes out as a tool for representing programs whose behaviour can be determined, at a certain time, by unpredictable events. In this case the choice has no guidance. Therefore the criterion of taking into account all possible cases when studying the convergency of the program (that is the total correctness criterion) is the most natural one.

We will analyze the distinction between the internal choice operator and the parallel synchronous operator using the logical distinction between disjunction and conjunction in section 5.

2.1 λ-calculus with Choice and Parallel Operators

Let $\Lambda_{+\|}$ be the set of pure λ-terms enriched with the binary operators $+$ and $\|$, that is the set of expressions generated by the following grammar:

$$M ::= x \mid \lambda x.M \mid MM \mid M + M \mid M\|M$$

where x ranges over a denumerable set *Var* of variables. As usual, $FV(M)$ is the set of variables which occur free in M. To simplify notation we assume that abstraction and application take precedence over $+$ and $\|$.

As usual, if $\longrightarrow_R$ is a one-step reduction relation on $\Lambda_{+\|}$, then $\xrightarrow{*}_R$ and $=_R$ denote the transitive and reflexive, the transitive and reflexive and symmetric closure of $\longrightarrow_R$, respectively. Finally $\xrightarrow{n}_R$ means the n-times self-composition of $\longrightarrow_R$.

To extend the β-reduction relation $\longrightarrow_\beta$ of classical λ-calculus to $\Lambda_{+\|}$, we explicitly mention rules (μ), (ν) and (ξ), instead of considering the closure under contexts of the β-rule. Therefore we implicitly forbid reductions of the form:

$$M \longrightarrow N \Rightarrow op(\ldots, M, \ldots) \longrightarrow op(\ldots, N, \ldots)$$

where op is either $+$ or $\|$.

We also define explicitly the subrelation of $\longrightarrow_\beta$ called in the literature *head reduction* (see [12] also for the subsequent notion of solvable terms).

Definition 1.

(i) The relation $\longrightarrow_\beta$ is the least binary relation on $\Lambda_{+\|}$ defined by:

$$(\beta)\ (\lambda x.M)N \longrightarrow_\beta M[N/x] \qquad (\mu)\ M \longrightarrow_\beta N \Rightarrow LM \longrightarrow_\beta LN$$

$$(\nu)\ M \longrightarrow_\beta N \Rightarrow ML \longrightarrow_\beta NL \quad (\xi)\ M \longrightarrow_\beta N \Rightarrow \lambda x.M \longrightarrow_\beta \lambda x.N.$$

(ii) The relation $\longrightarrow_\beta^h$ is the least binary relation on $\Lambda_{+\|}$ satisfying (β) and (ξ) above and

$$(\nu_\beta)\ M \longrightarrow_\beta^h M' \text{ and } M \notin \mathsf{Abst} \Rightarrow MN \longrightarrow_\beta^h M'N$$

where $\mathsf{Abst} = \{\lambda x.P \mid P \in \Lambda_{+\|}, x \in Var\}$.

In the solvability theory of the classical of λ-calculus, meaningful terms are not just those possessing a normal form with respect to $\longrightarrow_\beta$, but more in general those which determine a terminating $\longrightarrow_\beta$ reduction, when applied to suitable terms. These are characterized as those terms having a normal form with respect to the $\longrightarrow_\beta^h$ relation (see [12] Theorem 8.3.14). This normal form is called *head normal form*, and, in view of the characterization just mentioned, terms possessing a head normal form are called *solvable*.

Definition 2. The subset of $\Lambda_{+\|}$

$$\mathsf{SOL}_\beta = \{M \mid \exists M'.\ M \stackrel{*}{\longrightarrow}_\beta^h M' \text{ and } \neg\, \exists N.\ M' \longrightarrow_\beta^h N\}$$

is the set of β-solvable terms.

Note that $\longrightarrow_\beta^h$-reduction is deterministic since any term has at most one head redex because of rule (ν_β). Hence we have immediately:

$$M \in \mathsf{SOL}_\beta \quad \Leftrightarrow \quad \exists n\, \forall m \geq n.\ \neg\, \exists N.\ M \stackrel{m}{\longrightarrow}_\beta^h N.$$

2.2 The Parallel and Non-deterministic Calculus

In this subsection we think of $+$ as an *internal choice* operator and of $\|$ as a synchronous *parallel evaluator* of its arguments. Indeed, rule $(+_C)$ allows to freely choose between the arguments of $+$. Instead, $M\|N$ reduces according to rule $(\|_S)$ only when both M and N reduce. Moreover, since every term represents a function in the λ-calculus, we further define $M\|N$ as the function which, when applied to some L, returns $ML\|NL$ (rule $(\|_{app})$). All this is formalized in the following definition.

Definition 3.

(i) The relation $\longrightarrow_{pn}$ (*Parallel* and *Non-deterministic* reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying (β), (μ), (ν), (ξ) and

$$(+_c) \quad M + N \longrightarrow_{pn} M, \;\; M + N \longrightarrow_{pn} N$$
$$(\|_s) \quad M \longrightarrow_{pn} M', \; N \longrightarrow_{pn} N' \Rightarrow M\|N \longrightarrow_{pn} M'\|N'$$
$$(\|_{app}) \quad (M\|N)L \longrightarrow_{pn} ML\|NL.$$

(ii) The relation $\longrightarrow^h_{pn}$ (*Parallel* and *Non-deterministic* head reduction) is the least binary relation on $\Lambda_{+\|}$ satisfying (β), (ξ), $(+_c)$, $(\|_s)$, $(\|_{app})$ and

$$(\nu_{pn}) \;\; M \longrightarrow^h_{pn} M' \text{ and } M \notin \mathsf{Abst} \cup \mathsf{Par} \Rightarrow MN \longrightarrow^h_{pn} M'N$$

where $\mathsf{Par} = \{P\|Q \mid P, Q \in \Lambda_{+\|}\}$.

Because of rule $(+_c)$, the relation $\longrightarrow_{pn}$ is not confluent. Moreover, because of rule $(\|_s)$, the set of "head redexes" of a term M (that is the set of redexes that will be contracted in the first step of a $\longrightarrow^h_{pn}$ reduction) can be larger than a singleton. These facts imply that a term M may have more than one immediate reduct with respect to $\longrightarrow^h_{pn}$ (but always finitely many).

Consequently there are at least two natural ways of extending the notion of solvability to $\longrightarrow_{pn}$. We could say that M is solvable if at least one $\longrightarrow^h_{pn}$ reduction starting from M ends in a (head) normal form. This definition, however, does not distinguish between $+$ and $\|$ by the property of being solvable. Indeed, both $M + N$ and $M\|N$ would be solvable if and only if either M or N is solvable.

Since we are looking for a semantics keeping distinct $+$ and $\|$ wrt convergency, we define M to be solvable if and only if *all* head reductions starting from it terminate. We immediately have that $M + N$ is solvable if and only if both M *and* N are, while $M\|N$ is solvable if and only if *either* M is solvable *or* N is solvable.

As observed above, the reduction tree of any term under the relation $\longrightarrow^h_{pn}$ is a finitary tree, hence by König's Lemma, it is finite if and only if all its branches have finite lengths, i.e. there is an upper bound to the length of all head reductions of the given term. We use this in the following definition.

Definition 4. The subset of $\Lambda_{+\|}$

$$\mathsf{SOL}_{pn} = \{M \mid \exists n \, \forall m \geq n. \neg \exists N. \; M \xrightarrow{m}{}^h_{pn} N\}$$

is the set of pn-solvable terms.

As observed above, this definition of solvability fits well with the conjunctive behaviour of $+$ and the disjunctive behaviour of $\|$ since

$$M + N \in \mathsf{SOL}_{pn} \;\Leftrightarrow\; M \in \mathsf{SOL}_{pn} \text{ and } N \in \mathsf{SOL}_{pn}$$

while

$$M\|N \in \mathsf{SOL}_{pn} \;\Leftrightarrow\; M \in \mathsf{SOL}_{pn} \text{ or } N \in \mathsf{SOL}_{pn}.$$

For example, if $\mathbf{I} \equiv \lambda x.x$, and $\Delta \equiv \lambda x.xx$, we have that $\mathbf{I} + \Delta$ is pn-unsolvable, since $\mathbf{I} + \Delta\Delta \longrightarrow^h_{pn} \Delta\Delta \longrightarrow^h_{pn} \Delta\Delta$. Instead $\mathbf{I}\|\Delta\Delta$ is a normal form, so a fortiori it is pn-solvable. $\lambda x.(x\mathbf{I} + x(\Delta\Delta))$ is a pn-solvable term, since it head reduces to $\lambda x.x\mathbf{I}$ and to $\lambda x.x(\Delta\Delta)$. Notice that $\lambda x.x(\Delta\Delta)$ reduces to itself, but it is a head normal form.

2.3 Synchronous and Asynchronous Calculus

We introduce a slightly different reduction relation, still extending β-reduction and still ascribing a conjunctive semantics to $+$ and a disjunctive one to $||$. The aim is that of eliminating rule $(+_c)$. The advantage will be that the existence of a finite reduction path out of a term assures the solvability of the term (see Corollary 9). In this reduction $+$ is an *asynchronous* evaluator of its operands, while $||$ is a *synchronous* one. Moreover, both $+$ and $||$ have the feature of passing to their operands any argument to which they apply.

Definition 5.

(i) The relation $\longrightarrow_{sa}$ (*Synchronous* and *Asynchronous* reduction) is the least binary relation on $\Lambda_{+||}$ satisfying $(\beta), (\mu), (\nu), (\xi), (||_s), (||_{app})$ and

$$(+_a) \qquad M \longrightarrow_{sa} M' \Rightarrow \begin{cases} M+N \longrightarrow_{sa} M'+N \\ N+M \longrightarrow_{sa} N+M' \end{cases}$$

$$(+_{app}) \quad (M+N)L \longrightarrow_{sa} ML+NL.$$

(ii) The relation $\longrightarrow^h_{sa}$ (*Synchronous* and *Asynchronous* head reduction) is the least binary relation on $\Lambda_{+||}$ satisfying (β), (ξ), $(+_a)$, $(+_{app})$, $(||_s)$, $(||_{app})$ and

$$(\nu_{sa}) \quad M \longrightarrow^h_{sa} M' \text{ and } M \notin \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum} \Rightarrow MN \longrightarrow^h_{sa} M'N$$

where $\mathsf{Sum} = \{P+Q \mid P, Q \in \Lambda_{+||}\}$.

Even if rule $(+_c)$ has been dropped, the presence of rule $(+_a)$, together with the synchronous character of $||$, implies that $\longrightarrow_{sa}$ is not Church-Rosser. For example, being $\mathbf{I} \equiv \lambda x.x$, if $P \longrightarrow_{sa} P'$ and $Q \longrightarrow_{sa} Q'$, then $(P+Q)||\mathbf{II}$ reduces both to $(P'+Q)||\mathbf{I}$ and to $(P+Q')||\mathbf{I}$. These are normal forms, since the reducibility of a parallel composition requires reducibility of both its operands.

For the same reason the head reduction $\longrightarrow^h_{sa}$ is non-deterministic. Consequently, we define the notion of sa-solvability in the same way as we did for pn-solvability.

Definition 6. The subset of $\Lambda_{+||}$

$$\mathsf{SOL}_{sa} = \{M \mid \exists n\, \forall m \geq n.\, \neg\, \exists N.\, M \xrightarrow{m}{}^h_{sa} N\}$$

is the set of sa-solvable terms.

The difference between $+$ and $||$ with respect to the solvability criterion is still expressed as follows

$$M+N \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; M \in \mathsf{SOL}_{sa} \text{ and } N \in \mathsf{SOL}_{sa}$$

while

$$M||N \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; M \in \mathsf{SOL}_{sa} \text{ or } N \in \mathsf{SOL}_{sa}.$$

In spite of the lack of the Church-Rosser property, the existence of a finite $\longrightarrow^h_{sa}$-reduction path now implies the finiteness of all $\longrightarrow^h_{sa}$-reduction paths. To prove this we need to prove a more general statement, since a stronger induction hypothesis is used when dealing with rules (ξ) and (ν_{sa}). In particular, (ν_{sa}) forces us to consider term vectors and consequently rule (ξ) forces us to consider substitutions (see Proposition 8).

The following properties of the reduction relation $\longrightarrow^h_{sa}$ are crucial in subsequent proofs. They are an immediate consequence of the constraint in rule (ν_{sa}).

Proposition 7.

(i) *If* $P \in \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum}$, *then any head reduction out of* $PL_0\vec{L}$ *will start by reducing the subterm* PL_0.

(ii) *If* $P \equiv P_1$ *op* P_2 *(where op is* $+$ *or* $||$*) then any exhaustive head reduction of* $PL_0 \cdots L_{k-1}$ *will start with* k *steps leading to* $P_1L_0 \cdots L_{k-1}$ *op* $P_2L_0 \cdots L_{k-1}$.

As usual a substitution is a map from variables to terms which is the identity for all variables but a finite set.

Proposition 8. *If* $M \longrightarrow^h_{sa} N$, *then*

$$\forall (\cdot)^\nabla, \vec{L}.\ N^\nabla \vec{L} \in \mathsf{SOL}_{sa} \ \Leftrightarrow\ M^\nabla \vec{L} \in \mathsf{SOL}_{sa},$$

where $(\cdot)^\nabla$ *ranges over substitutions and* $\vec{L}$ *is a vector of terms.*

Proof. By induction on the definition of $\longrightarrow^h_{sa}$.

Case $(+_a)$: then $M \equiv P + Q \longrightarrow^h_{sa} P' + Q \equiv N$ with $P \longrightarrow^h_{sa} P'$. Now

$$\begin{aligned}(P'^\nabla + Q^\nabla)\vec{L} \in \mathsf{SOL}_{sa} &\Leftrightarrow P'^\nabla\vec{L} + Q^\nabla\vec{L} \in \mathsf{SOL}_{sa}\\ &\Leftrightarrow P'^\nabla\vec{L} \in \mathsf{SOL}_{sa} \text{ and } Q^\nabla\vec{L} \in \mathsf{SOL}_{sa}\\ &\Leftrightarrow P^\nabla\vec{L} \in \mathsf{SOL}_{sa} \text{ and } Q^\nabla\vec{L} \in \mathsf{SOL}_{sa} \quad \text{by induction}\\ &\Leftrightarrow P^\nabla\vec{L} + Q^\nabla\vec{L} \in \mathsf{SOL}_{sa}\\ &\Leftrightarrow (P^\nabla + Q^\nabla)\vec{L} \in \mathsf{SOL}_{sa}\end{aligned}$$

where the $\Leftarrow$ part of the first implication and the last $\Leftrightarrow$ are trivial if the vector $\vec{L}$ is empty. Otherwise they readily follows from 7*(i)*.

Case $(+_{app})$: then $M \equiv (P + Q)R \longrightarrow^h_{sa} PR + QR \equiv N$. We have:

$$(P^\nabla R^\nabla + Q^\nabla R^\nabla)\vec{L} \in \mathsf{SOL}_{sa} \ \Leftrightarrow\ (P^\nabla + Q^\nabla)R^\nabla\vec{L} \in \mathsf{SOL}_{sa}$$

as in previous case.

Case $(||_s)$: then $M \equiv P||Q \longrightarrow^h_{sa} P'||Q' \equiv N$ with $P \longrightarrow^h_{sa} P'$ and $Q \longrightarrow^h_{sa} Q'$. Then this case is similar to case $(+_a)$, where $+$ is replaced by $||$ and "$\ldots \in \mathsf{SOL}_{sa}$ and $\ldots \in \mathsf{SOL}_{sa}$" is replaced by "$\ldots \in \mathsf{SOL}_{sa}$ or $\ldots \in \mathsf{SOL}_{sa}$".

Case $(||_{app})$: same as case $(+_{app})$ where $+$ is replaced by $||$.

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{sa} P[Q/x] \equiv N$. By 7*(i)*, the first step out of $(\lambda x.P^\nabla)Q^\nabla\vec{L}$ must be a β-reduction. Then

$$M^\nabla\vec{L} \equiv (\lambda x.P^\nabla)Q^\nabla\vec{L} \in \mathsf{SOL}_{sa} \ \Leftrightarrow\ N^\nabla\vec{L} \equiv P^\nabla[Q^\nabla/x]\vec{L} \in \mathsf{SOL}_{sa}.$$

Note that, being x bound in $\lambda x.P$, we can freely assume that the substitution $(\cdot)^\nabla$ does not affect it.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{sa} \lambda x.P' \equiv N$ with $P \longrightarrow^h_{sa} P'$. If the vector $\vec{L}$ is empty, then the thesis follows from the induction hypothesis. Otherwise, taking the non empty vector $L_0\vec{L}$, the first step out of $(\lambda x.P'^\nabla)L_0\vec{L}$ will be a β-reduction by 7*(i)*. Then:

$$\begin{aligned}(\lambda x.P'^\nabla)L_0\vec{L} \in \mathsf{SOL}_{sa} &\Leftrightarrow (P'^\nabla[L_0/x])\vec{L} \in \mathsf{SOL}_{sa}\\ &\Leftrightarrow (P^\nabla[L_0/x])\vec{L} \in \mathsf{SOL}_{sa} \quad \text{by induction}\\ &\Leftrightarrow (\lambda x.P^\nabla)L_0\vec{L} \in \mathsf{SOL}_{sa},\end{aligned}$$

where in the induction hypothesis the substitution is the composition of $(\cdot)^\nabla$ and $[L_0/x]$. As in case (β) we assume that $(\cdot)^\nabla$ does not substitute for x.

Case (ν_{sa}): then $M \equiv PQ \longrightarrow^h_{sa} P'Q \equiv N$ with $P \longrightarrow^h_{sa} P'$. Then, by the induction hypothesis, taking the vector $Q^\nabla \vec{L}$, we immediately have that

$$N^\nabla \vec{L} \equiv P'^\nabla Q^\nabla \vec{L} \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; P^\nabla Q^\nabla \vec{L} \equiv M^\nabla \vec{L} \in \mathsf{SOL}_{sa}.$$

□

Corollary 9. *$M \in \mathsf{SOL}_{sa} \;\Leftrightarrow\; \exists M'.\, M \xrightarrow{*}{}^h_{sa} M'$ and $\neg\,\exists N.\, M' \longrightarrow^h_{sa} N$.*

Proof. $\Rightarrow$ is trivial.
The proof of $\Leftarrow$ follows by straightforward induction on the length of the reduction $M \xrightarrow{*}{}^h_{sa} M'$ using Proposition 8 with the identical substitution and the empty vector. □

2.4 Relationships between the two Calculi

Even if the reductions $\longrightarrow^h_{pn}$ and $\longrightarrow^h_{sa}$ are different, as it is clear also from Corollary 9, they are equivalent in the sense that they determine the same set of solvable terms, i.e. SOL_{pn} and SOL_{sa} coincide.

To show this we need a definition and some Lemmas, all proved by induction on the structure of one-step head reductions.

Definition 10. Define SOL^n_{sa} as the set of terms whose longest $\longrightarrow^h_{sa}$ reduction has at most n steps, i.e.:

$$\mathsf{SOL}^n_{sa} = \{M \mid \forall m \geq n.\, \neg\,\exists N.\, M \xrightarrow{m}{}^h_{sa} N\}.$$

Comparing this with Definition 6 it is clear that $\mathsf{SOL}_{sa} = \bigcup_{n\geq 0} \mathsf{SOL}^n_{sa}$.

The first lemma connects the reduction $\longrightarrow^h_{pn}$ with the set SOL^n_{sa}.

Lemma 11. *If $M \longrightarrow^h_{pn} N$ then, for all $\vec{L}$ and substitutions $(\cdot)^\nabla$:*

$$M^\nabla \vec{L} \in \mathsf{SOL}^n_{sa} \Rightarrow \exists m \leq n.\, N^\nabla \vec{L} \in \mathsf{SOL}^m_{sa}.$$

Moreover, if $m = n$, then we used rule $(+_c)$ in deriving that $M \longrightarrow^h_{pn} N$.

Proof. By induction on $\longrightarrow^h_{pn}$.

Case $(+_c)$: then $M \equiv P + Q \longrightarrow^h_{pn} P \equiv N$, say.
If $(P{+}Q)^\nabla \vec{L} \equiv (P^\nabla {+} Q^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa}$ and r is the length of $\vec{L}$, then by 7*(ii)* any $\longrightarrow^h_{sa}$ reduction out of $(P^\nabla + Q^\nabla)\vec{L}$ will produce $P^\nabla \vec{L} + Q^\nabla \vec{L}$ in r steps. Hence $P^\nabla \vec{L} + Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa}$ and, a fortiori, $P^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa}$. If $\vec{L}$ is empty, we get $m = n$.

Case $(||_s)$: then $M \equiv P||Q \longrightarrow^h_{pn} P'||Q' \equiv N$ with $P \longrightarrow^h_{pn} P'$ and $Q \longrightarrow^h_{pn} Q'$.
Now if r is the length of $\vec{L}$, then

$$\begin{aligned} M^\nabla \vec{L} \equiv (P^\nabla || Q^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa} &\Rightarrow P^\nabla \vec{L} || Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \\ &\Rightarrow P^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \text{ or } Q^\nabla \vec{L} \in \mathsf{SOL}^{n-r}_{sa} \\ &\Rightarrow \exists m \leq n-r.\, P'^\nabla \vec{L} \in \mathsf{SOL}^m_{sa} \text{ or } Q'^\nabla \vec{L} \in \mathsf{SOL}^m_{sa} \\ &\qquad\qquad\qquad\qquad\text{by induction} \\ &\Rightarrow \exists m \leq n-r.\, (P'^\nabla || Q'^\nabla)\vec{L} \equiv N^\nabla \vec{L} \in \mathsf{SOL}^{m+r}_{sa} \end{aligned}$$

and clearly $m + r \leq n$.

Notice that if $\vec{L}$ is empty, we can have $m = n$. In this case $P^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$ or $Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$. So we have by induction that we used rule $(+_c)$ in deriving $P \longrightarrow^h_{pn} P'$ or $Q \longrightarrow^h_{pn} Q'$. Therefore rule $(+_c)$ has also been used in deriving $M \longrightarrow^h_{pn} N$.

Case $(\|_{app})$: then $M \equiv (P\|Q)R \longrightarrow^h_{pn} PR\|QR \equiv N$.

If $((P\|Q)R)^\nabla \vec{L} \equiv (P^\nabla\|Q^\nabla)R^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then we immediately have

$$(P^\nabla R^\nabla \| Q^\nabla R^\nabla)\vec{L} \equiv (PR\|QR)^\nabla \vec{L} \in \mathsf{SOL}^{n-1}_{sa}.$$

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{pn} P[Q/x] \equiv N$.

Now for all $(\cdot)^\nabla$ $((\lambda x.P)Q)^\nabla \equiv (\lambda x.P^\nabla)Q^\nabla$ up to renaming of the bound variable x, and for all $\vec{L}$, any $\longrightarrow^h_{sa}$ reduction out of $(\lambda x.P^\nabla)Q^\nabla \vec{L}$ will start by

$$(\lambda x.P^\nabla)Q^\nabla \vec{L} \longrightarrow^h_{sa} P^\nabla[Q^\nabla/x]\vec{L}$$

hence, if $(\lambda x.P^\nabla)Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then $P^\nabla[Q^\nabla/x]\vec{L} \in \mathsf{SOL}^{n-1}_{sa}$.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{pn} \lambda x.P' \equiv N$, with $P \longrightarrow^h_{pn} P'$.

Now, up to renaming of the bound variable x, $(\lambda x.P)^\nabla \equiv \lambda x.P^\nabla$. Assume that $(\lambda x.P^\nabla)\vec{L} \in \mathsf{SOL}^n_{sa}$, then if $\vec{L}$ is empty the thesis follows immediately by induction. Otherwise the first step of any $\longrightarrow^h_{sa}$ will be

$$(\lambda x.P^\nabla)Q\vec{L} \longrightarrow^h_{sa} P^\nabla[Q/x]\vec{L},$$

so that $P^\nabla[Q/x]\vec{L} \in \mathsf{SOL}^{n-1}_{sa}$. From the induction hypothesis there exists $m \leq n-1$ such that

$$P'^\nabla[Q/x]\vec{L} \in \mathsf{SOL}^m_{sa}$$

which implies that

$$(\lambda x.P'^\nabla)Q\vec{L} \in \mathsf{SOL}^{m+1}_{sa}$$

and clearly $m + 1 \leq n$.

Case (ν_{pn}): then $M \equiv PQ \longrightarrow^h_{pn} P'Q \equiv N$ with $P \longrightarrow^h_{pn} P'$, where $P \notin \mathsf{Abst} \cup \mathsf{Par}$.

Now, if $(PQ)^\nabla \vec{L} \equiv P^\nabla Q^\nabla \vec{L} \in \mathsf{SOL}^n_{sa}$, then by induction and considering the vector $Q^\nabla \vec{L}$ we have $P'^\nabla Q^\nabla \vec{L} \in \mathsf{SOL}^m_{sa}$ for some $m \leq n$ and we are done.

If $m = n$, then by induction we used rule $(+_c)$ in deriving $P \longrightarrow^h_{pn} P'$. Therefore we used rule $(+_c)$ also in deriving $M \longrightarrow^h_{pn} N$. □

Lemma 12. *If* $M \longrightarrow^h_{sa} M'$ *then, for all* $\vec{L}$ *and substitution* $(\cdot)^\nabla$*:*

$$M'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow \exists N.\ N^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } M \longrightarrow^h_{pn} N.$$

Proof. By induction on $\longrightarrow^h_{sa}$.

Case $(+_a)$: then assume that $M \equiv P + Q \longrightarrow^h_{sa} P' + Q \equiv M'$ with $P \longrightarrow^h_{sa} P'$. Now

$$(P'^\nabla + Q^\nabla)\vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ or } Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa}.$$

If $P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, choosing $N \equiv P$, we have

$$M \equiv P + Q \longrightarrow^h_{pn} N$$

and by Proposition 8

$$P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P^\nabla \vec{L} \notin \mathsf{SOL}_{sa}.$$

Otherwise, if $Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, we take $N \equiv Q$ and we have $M \equiv P + Q \longrightarrow^h_{pn} N$. The case $M \equiv P + Q \longrightarrow^h_{sa} P + Q' \equiv M'$, with $Q \longrightarrow^h_{sa} Q'$, is symmetric.

Case $(+_{app})$: then $M \equiv (P+Q)R \longrightarrow^h_{sa} PR + QR \equiv M'$. Now

$$(P^\nabla R^\nabla + Q^\nabla R^\nabla)\vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow P^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ or } Q^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa} .$$

If $P^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, then it suffices to choose $N \equiv PR$, with $(P+Q)R \longrightarrow^h_{pn} PR$. Otherwise $Q^\nabla R^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, so that we choose $N \equiv QR$ and we conclude similarly.

Case $(||_s)$: then $M \equiv P||Q \longrightarrow^h_{sa} P'||Q' \equiv M'$ with $P \longrightarrow^h_{sa} P'$ and $Q \longrightarrow^h_{sa} Q'$. If $(P'^\nabla || Q'^\nabla)\vec{L} \notin \mathsf{SOL}_{sa}$, then both $P'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$ and $Q'^\nabla \vec{L} \notin \mathsf{SOL}_{sa}$, so that, by induction

$$\exists N_1, N_2.\ N_1^\nabla \vec{L}, N_2^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow^h_{pn} N_1 \text{ and } Q \longrightarrow^h_{pn} N_2 .$$

Therefore we choose $N \equiv N_1||N_2$.

Case $(||_{app})$: then $M \equiv (P||Q)R \longrightarrow^h_{sa} PR||QR \equiv M'$. Hence we take $N \equiv M'$.

Case (β): then $M \equiv (\lambda x.P)Q \longrightarrow^h_{sa} P[Q/x] \equiv M'$. Clearly the choice $N \equiv M'$ works.

Case (ξ): then $M \equiv \lambda x.P \longrightarrow^h_{sa} \lambda x.P' \equiv M'$ with $P \longrightarrow^h_{sa} P'$.

If the vector $\vec{L}$ is empty, then the thesis follows from the induction hypothesis. Otherwise consider the non empty vector $L_0\vec{L}$:

$$\begin{aligned}(\lambda x.P'^\nabla)L_0\vec{L} \notin \mathsf{SOL}_{sa} &\Rightarrow P'^\nabla[L_0/x]\vec{L} \notin \mathsf{SOL}_{sa} && \text{by 7}(i)\\ &\Rightarrow \exists N'.\ N'^\nabla[L_0/x]\vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow^h_{pn} N' && \text{by induction.}\end{aligned}$$

Then $\lambda x.P \longrightarrow^h_{pn} \lambda x.N'$ and we take $N \equiv \lambda x.N'$.

Case (ν_{sa}): then $M \equiv PQ \longrightarrow^h_{sa} P'Q \equiv M'$ with $P \longrightarrow^h_{sa} P'$ and $P \notin \mathsf{Abst} \cup \mathsf{Par} \cup \mathsf{Sum}$. From the induction hypothesis

$$P'^\nabla Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \Rightarrow \exists N'.\ N'^\nabla Q^\nabla \vec{L} \notin \mathsf{SOL}_{sa} \text{ and } P \longrightarrow^h_{pn} N' .$$

Then $PQ \longrightarrow^h_{pn} N'Q$ by (ν_{pn}) since in particular $P \notin \mathsf{Abst} \cup \mathsf{Par}$. Therefore we take $N \equiv N'Q$.

□

We are now ready to prove that $\longrightarrow^h_{pn}$ and $\longrightarrow^h_{sa}$ determine the same set of solvable terms. To prove this, we will apply the previous Lemmas, using the identical substitution and the empty vector of terms.

Theorem 13. $\mathsf{SOL}_{sa} = \mathsf{SOL}_{pn}$.

Proof. First we show that $\mathsf{SOL}_{sa} \subseteq \mathsf{SOL}_{pn}$. Toward a contradiction suppose that $M \in \mathsf{SOL}_{sa}$ but $M \notin \mathsf{SOL}_{pn}$. If $M \in \mathsf{SOL}_{sa}$, then there exists n such that $M \in \mathsf{SOL}^n_{sa}$. The hypothesis that $M \notin \mathsf{SOL}_{pn}$ implies that there is a set $\{M_i\}_{i \in \omega}$ such that $M_0 \equiv M$ and, for all i, $M_i \longrightarrow^h_{pn} M_{i+1}$. By Lemma 11 there is a k such that $M_k \in \mathsf{SOL}^0_{sa}$, i.e. M_k is in normal form wrt $\longrightarrow^h_{sa}$. This is because the only case in which the n of SOL^n_{sa} does not decrease is when in the $\longrightarrow^h_{pn}$ reduction rule $(+_c)$ is used. But the number of consecutive steps of this kind is bounded by the number of the occurrences of $+$ in the term to be reduced.
It is easy to see that, if M_k can be further reduced under $\longrightarrow^h_{pn}$, then only steps involving the use of $(+_c)$ are possible, which again are bounded by the number of $+$'s in M_k. So any sequence of $\longrightarrow^h_{pn}$ reductions out of M has to be finite: a contradiction.

To show that $\mathsf{SOL}_{pn} \subseteq \mathsf{SOL}_{sa}$ assume, toward a contradiction, that $M \in \mathsf{SOL}_{pn}$ and $M \notin \mathsf{SOL}_{sa}$. Then there exists M_1 such that $M \longrightarrow^h_{sa} M_1$ and $M_1 \notin \mathsf{SOL}_{sa}$. By Lemma 12 this implies that there exists N such that $M \longrightarrow^h_{pn} N$ and still $N \notin \mathsf{SOL}_{sa}$. Iterating the same reasoning, we build an infinite $\longrightarrow^h_{pn}$ reduction out of M, so that $M \notin \mathsf{SOL}_{pn}$: a contradiction. □

Since our aim is that of developing an unfolding semantics for our calculus, we are interested essentially in the set of solvable terms. So Theorem 13 gives us the possibility of choosing freely between the reduction relations $\longrightarrow_{pn}$ and $\longrightarrow_{sa}$. For technical reasons we will concentrate in the following on $\longrightarrow_{sa}$. Consequently we will write simply $\longrightarrow$ for it, and SOL for the set of solvable terms.

3 Operational Semantics

In the previous section the semantics of our calculi has been described by means of reduction relations. Here we develop a theory to compare terms with respect to their functional behaviours. We do this in two different ways. The first one is by means of contexts. The second one is more refined and compares terms by means of their "approximants", where the set of approximants of a term can be viewed as a generalization to our calculus of the notion of Böhm tree.

3.1 Contextual Semantics

Following the standard approach for defining equational theories from convergency predicates (originated with Morris' thesis [33]; see also [12] 16.5.5), we state:

Definition 14. For any $M, N \in \Lambda_{+||}$ we define:

$$M \sqsubseteq^{\mathcal{O}} N \Leftrightarrow \forall C[\,].\ C[M] \in \mathsf{SOL} \Rightarrow C[N] \in \mathsf{SOL}.$$

Accordingly,

$$M \simeq^{\mathcal{O}} N \Leftrightarrow M \sqsubseteq^{\mathcal{O}} N \sqsubseteq^{\mathcal{O}} M.$$

Clearly, the relation $\sqsubseteq^{\mathcal{O}}$ is a precongruence. The set SOL, when restricted to pure λ-terms, is the set of terms having a head normal form, that is those terms which are solvable in the classical sense. Hence the restriction of $\simeq^{\mathcal{O}}$ to pure λ-terms is the λ-theory of D_∞ by a well known result of Wadsworth [45].

Proposition 15. *The following (in)-equations hold:*

(i) $(\lambda x.M)N \simeq^{\mathcal{O}} M[N/x]$;
(ii) $(M+N)L \simeq^{\mathcal{O}} ML + NL$;
(iii) $L(M+N) \sqsubseteq^{\mathcal{O}} LM + LN$;
(iv) $(M||N)L \simeq^{\mathcal{O}} ML||NL$;
(v) $LM||LN \sqsubseteq^{\mathcal{O}} L(M||N)$;
(vi) $\lambda x.(M+N) \simeq^{\mathcal{O}} \lambda x.M + \lambda x.N$;
(vii) $\lambda x.(M||N) \simeq^{\mathcal{O}} \lambda x.M||\lambda x.N$;
(viii) $M+N \sqsubseteq^{\mathcal{O}} M, N$;
(ix) $L \sqsubseteq^{\mathcal{O}} M, N \Rightarrow L \sqsubseteq^{\mathcal{O}} M+N$;
(x) $M, N \sqsubseteq^{\mathcal{O}} M||N$;
(xi) $M, N \sqsubseteq^{\mathcal{O}} L \Rightarrow M||N \sqsubseteq^{\mathcal{O}} L$.

where the inequalities (iii) and (v) are in general proper.

Proof. We consider only the interesting cases.

To prove that the inequality *(iii)* is proper, let $\Delta \equiv \lambda x.xx$, $M \equiv \lambda x.x(\lambda yzv.v)\Delta$ and $N \equiv \lambda x.\Delta$. ΔM and ΔN both β-reduce to Δ and therefore $\Delta M + \Delta N$ is solvable. Instead, $\Delta(M+N)$ reduces to $\Delta + \Delta\Delta + \Delta + \Delta$, which is unsolvable.

To prove that the inequality *(v)* is proper, let Δ be as above, $\mathbf{I} \equiv \lambda x.x$, $\mathbf{K} \equiv \lambda xy.x$, $T \equiv \lambda x.x\Delta\mathbf{I}\Delta$ and $R \equiv \lambda x.x\Delta\Delta$. $(T+R)(\mathbf{I}||\mathbf{K})$ is solvable since it reduces to $(\Delta||\Delta\Delta) + (\Delta\Delta||\Delta)$. Instead, $(T+R)\mathbf{I}||(T+R)\mathbf{K}$ reduces to $(\Delta + \Delta\Delta)||(\Delta\Delta + \Delta)$ and therefore it is unsolvable.

(ix). First, we prove the idempotence of $+$. $P + P \sqsubseteq^{\mathcal{O}} P$ follows immediately from *(viii)*. $P \sqsubseteq^{\mathcal{O}} P + P$ follows from *(iii)* choosing $L \equiv \mathbf{K}P$. Now, given an arbitrary context $C[\ ]$, let $C'[\ ] \equiv C[[\ ] + L]$ and $C''[\ ] \equiv C[M + [\ ]]$. If $L \sqsubseteq^{\mathcal{O}} M, N$, then

$$C[L] \in \mathsf{SOL} \Rightarrow C[L + L] \equiv C'[L] \in \mathsf{SOL} \Rightarrow C'[M] \equiv C''[L] \in \mathsf{SOL}$$
$$\Rightarrow C''[N] \equiv C[M + N] \in \mathsf{SOL}.$$

(xi). Similarly, we prove the idempotence of $||$ using *(x)* and *(v)*. Now, given an arbitrary context $C[\ ]$, let $C'[\ ] \equiv C[[\ ]||N]$ and $C''[\ ] \equiv C[L||[\ ]]$. If $M, N \sqsubseteq^{\mathcal{O}} L$, then

$$C[M||N] \equiv C'[M] \in \mathsf{SOL} \Rightarrow C'[L] \equiv C''[N] \in \mathsf{SOL} \Rightarrow C''[L] \equiv C[L||L] \in \mathsf{SOL}$$
$$\Rightarrow C[L] \in \mathsf{SOL}.$$

□

3.2 Capabilities Semantics

$\simeq^{\mathcal{O}}$ is an extensional theory by definition, and in fact $\lambda x.(M + N) \simeq^{\mathcal{O}} \lambda x.M + \lambda x.N$ holds. However, if $+$ is interpreted as an operation to form "sets" of values and λx is the standard functional abstraction, then this equality identifies any set of functions with a single multivalued function (see [30, 29]). This is not very natural if one considers that $L(M+N) \not\simeq^{\mathcal{O}} LM + LN$. This problem becomes more evident when modeling the calculus by means of type assignment systems, as we shall do in the forthcoming sections.

For these reasons we introduce a finer, non extensional semantics which is still based on the notion of head normal form and solvability, but uses ideas underlining Böhm trees. More precisely, we first show the shape of head normal forms in the present setting. Then we associate to each term the set of head normal forms (the *capabilities*) which can be obtained out of it using a more liberal reduction relation ($\longrightarrow_a$, see Definition 19). Lastly we define a notion of approximation patterned after [45] and we compare terms via the approximate normal forms of their capabilities.

It is easy to verify that the terms irreducible according to $\longrightarrow^h$ (i.e. the head normal forms) satisfy the conditions of the following proposition.

Proposition 16. *The set $\mathcal{H}$ of* head normal forms *is the least one such that:*

(a) $M_1, \ldots, M_n \in \Lambda_{+||},\ x \in Var \ \Rightarrow\ xM_1 \ldots M_n \in \mathcal{H}\ \ (n \geq 0)$*;*
(b) $H \in \mathcal{H}, x \in Var \ \Rightarrow\ \lambda x.H \in \mathcal{H}$*;*
(c) $H_1, H_2 \in \mathcal{H} \ \Rightarrow\ H_1 + H_2 \in \mathcal{H}$*;*
(d) $H \in \mathcal{H}, M \in \Lambda_{+||} \ \Rightarrow\ H||M, M||H \in \mathcal{H}$*.*

Definition 17. The set $\mathcal{H}(M)$ of *head normal forms* of M is defined by:

$$\mathcal{H}(M)\ =\ \{H \in \mathcal{H} \mid M \xrightarrow{*}{}^{h} H\}.$$

For example, let us consider the terms $F\mathsf{0}$ and $G\mathsf{0}$, where

$$F \equiv \Theta(\lambda fx.(x + f(\mathsf{Succ}\ x))),\ \ G \equiv \Theta(\lambda fx.(x||f(\mathsf{Succ}\ x))),$$

$\Theta \equiv (\lambda zx.x(zzx))(\lambda zx.x(zzx))$ is the Turing fixed point combinator, $\mathsf{0}$ and Succ are the zero and successor of Church numerals respectively. Let n be the Church numeral for the natural number n, then it is easy to check that for any n

$$F\mathsf{0} \xrightarrow{*}{}^{h} \mathsf{0} + \mathsf{1} + \ldots + \mathsf{n} + F(\mathsf{Succ}\ \mathsf{n})$$

which is never in $\mathcal{H}$. So $\mathcal{H}(F0) = \emptyset$.
On the other hand $\mathcal{H}(G0) = \{0 \| G(\mathsf{Succ}\ 0)\}$. However, if we consider its reducts with respect to $\longrightarrow$, then we see that for any n, putting $G' \equiv (\lambda f x.(x \| f(\mathsf{Succ}\ x)))$, we have:

$$\begin{aligned} G0 &\xrightarrow{*} G'G0 \\ &\cdots \\ &\xrightarrow{*} \underbrace{G'(\cdots(G'}_{n+1}G)\cdots)0 \\ &\xrightarrow{*} 0 \| \underbrace{G'(\cdots(G'}_{n}G)\cdots)1 \end{aligned}$$

giving rise to an infinite set of (distinct) head normal forms, none of which even reduces to a head normal form of the shape

$$0 \| 1 \| \ldots \| \mathsf{n} \| G(\mathsf{Succ}\ \mathsf{n}),$$

because of the synchronous character of $\|$. This is unfortunate, since the last term is a better candidate for describing the behaviour of $G0$ when it is applied to an argument.

Being $\mathcal{H}$ the set of normal forms wrt $\longrightarrow^h$, by Corollary 9, it follows that

$$\mathsf{SOL} = \{M \in \Lambda_{+\|} \mid \mathcal{H}(M) \neq \emptyset\}.$$

Observe that $H \in \mathcal{H}(M+N)$ implies $H \equiv H_1 + H_2$ where $H_1 \in \mathcal{H}(M)$ and $H_2 \in \mathcal{H}(N)$, while $H \in \mathcal{H}(M \| N)$ implies $H \equiv L_1 \| L_2$ where $L_1 \in \mathcal{H}(M)$ or $L_2 \in \mathcal{H}(N)$, only.

Remark 18. Since $\longrightarrow^h \subseteq \longrightarrow$, it holds

$$\mathsf{SOL} \subseteq \{M \in \Lambda_{+\|} \mid \exists H \in \mathcal{H}.\ M \xrightarrow{*} H\}.$$

Also the viceversa is true, because, by a standardization argument, $M \xrightarrow{*} H$ implies that

$$\exists N.\ M \xrightarrow{*}^h N \text{ and } N \xrightarrow{*}^i H$$

where $\longrightarrow^i$ is obtained out of $\longrightarrow$ by forbidding the $\longrightarrow^h$ steps. In other words, only internal redexes are reduced according to $\longrightarrow^i$. But we omit this quite long proof, since we do not need this result.
Notice that, due to the lack of the Church-Rosser property, our language does not fit the conditions of [21], therefore we cannot directly use their proof method.

As it is clear from Proposition 16, we have shifted to the head normal forms the distinction between the conjunctive behaviour of $+$ and the disjunctive nature of $\|$. We capitalize on this fact and we remedy the drawback outlined in the above example by abstracting away from the synchronous reduction of $\|$.

Definition 19.

(i) Let $\longrightarrow_a$ be the least binary relation on $\Lambda_{+\|}$ which is defined as $\longrightarrow$ adding the clause:

$$M \longrightarrow_a M' \Rightarrow M \| N \longrightarrow_a M' \| N \text{ and } N \| M \longrightarrow_a N \| M'.$$

(ii) The set $\mathcal{C}(M)$ of the capabilities of M is defined by:

$$\mathcal{C}(M) \ = \ \{H \mid \exists H' \in \mathcal{H}(M).\ H' \xrightarrow{*}_a H\}.$$

As examples, consider the terms $F0$ and $G0$ and observe that $\mathcal{C}(F0) = \emptyset$, while

$$0||1||\dots||\mathsf{n}||G(\mathsf{Succ}\ \mathsf{n}) \in \mathcal{C}(G0) \text{ for all } n \geq 0.$$

We now introduce the formal definition of approximate normal form. This will be useful for comparing the capabilities of terms through their approximate normal forms (see Definition 23).

Definition 20. Let $\Lambda_{+||\Omega}$ be the language obtained from $\Lambda_{+||}$ by adding the constant Ω. The set of *approximate normal forms* $\mathcal{A} \subset \Lambda_{+||\Omega}$ is the least one such that:

(i) $\Omega \in \mathcal{A}$;

(ii) $A_1, \dots, A_n \in \mathcal{A} \Rightarrow xA_1 \dots A_n \in \mathcal{A}\ \ (n \geq 0)$;

(iii) $A \in \mathcal{A} \Rightarrow \lambda x.A \in \mathcal{A}$;

(iv) $A_1, A_2 \in \mathcal{A} \Rightarrow A_1 + A_2, A_1||A_2 \in \mathcal{A}$.

We define a preorder relation on approximate normal forms which generalizes the classical one taking into account the intended meanings of $+$ and $||$. Moreover an η-redex is always less than its contractum according to this preorder.

Definition 21. Over the set $\mathcal{A}$ define $\preceq$ as the least preorder which makes $\mathcal{A}$ into a distributive lattice with $+$ as meet, $||$ as join and Ω as bottom, and such that:

(i) $\lambda x.\Omega \preceq \Omega$;

(ii) $A \preceq A' \Rightarrow \lambda x.A \preceq \lambda x.A'$;

(iii) $A_1 \preceq A'_1, \dots, A_n \preceq A'_n \Rightarrow xA_1 \dots A_n \preceq xA'_1 \dots A'_n$;

(iv) $\lambda x.(A||A') \preceq \lambda x.A||\lambda x.A'$;

(v) $\lambda y.xA_1 \dots A_n y \preceq xA_1 \dots A_n$, if $y \notin FV(xA_1 \dots A_n)$.

Let $\sim$ be the equivalence relation induced by $\preceq$.

As usual, we associate to each term M an approximate normal form $\phi(M)$ obtained by replacing Ω to all subterms which are not head normal forms.

Definition 22. Let $\phi: \Lambda_{+||} \to \mathcal{A}$ be the following map:

(i) $\phi(\lambda x_1 \dots x_n.xM_1 \dots M_m) = \lambda x_1 \dots x_n.x\phi(M_1) \dots \phi(M_m)$;

(ii) $\phi(\lambda x_1 \dots x_n.H + H') = \lambda x_1 \dots x_n.\phi(H) + \phi(H')$, if $H, H' \in \mathcal{H}$;

(iii) $\left.\begin{array}{l}\phi(\lambda x_1 \dots x_n.M||H) = \lambda x_1 \dots x_n.\phi(M)||\phi(H)\\ \phi(\lambda x_1 \dots x_n.H||M) = \lambda x_1 \dots x_n.\phi(H)||\phi(M)\end{array}\right\}$ if $H \in \mathcal{H}$;

(iv) $\phi(M) = \Omega$, if $M \notin \mathcal{H}$.

Now we relate the capabilities of two terms by comparing their approximate normal forms in a cofinal way.

Definition 23. For any $M, N \in \Lambda_{+||}$ we define:

$$M \sqsubseteq^{\mathcal{A}} N \Leftrightarrow \forall H \in \mathcal{C}(M)\ \exists H' \in \mathcal{C}(N).\ \phi(H) \preceq \phi(H').$$

Accordingly,

$$M \simeq^{\mathcal{A}} N \Leftrightarrow M \sqsubseteq^{\mathcal{A}} N \sqsubseteq^{\mathcal{A}} M.$$

The possibility of taking an element out of the set of capabilities allows us to choose any term obtainable by reducing according to $\longrightarrow_a$. Notice that $\longrightarrow_a$ is the more permissive among the reduction relation we introduced. The fact of considering then the approximate normal form of this term means (as usual) that we disregard redexes.

If one defines the set of approximants of a term as the downward closure of the set of approximate normal forms of its capabilities, one immediately obtains that the relation $\sqsubseteq^{\mathcal{A}}$ coincides with the inclusion between sets of approximants.

Definition 24. Let $M \in \Lambda_{+\|}$, then the *set* $\mathcal{A}(M)$ *of approximants of* M is defined by:

$$\mathcal{A}(M) = \{A \in \mathcal{A} \mid \exists H \in \mathcal{C}(M).\ A \preceq \phi(H)\} \cup \{\Omega\}.$$

For example,

$$\mathsf{0}\|\mathsf{1}\|\ldots\|\mathsf{n}\|\Omega \in \mathcal{A}(G\mathsf{0}) \text{ for all } n \geq 0.$$

The following properties of the sets of approximants follow immediately from previous definitions.

Proposition 25.

(i) $\mathcal{A}(M+N) = \mathcal{A}(M) \cap \mathcal{A}(N)$;
(ii) $\mathcal{A}(M\|N) = \{H\|H' \mid H \in \mathcal{A}(M) \text{ and } H' \in \mathcal{A}(N)\}$;
(iii) $M \sqsubseteq^{\mathcal{A}} N \Leftrightarrow \mathcal{A}(M) \subseteq \mathcal{A}(N)$;
(iv) $M \xrightarrow{*}{}^{h} N \Rightarrow \mathcal{A}(N) \subseteq \mathcal{A}(M)$.

Remark 26. 25 *(iv)* is weak. Indeed a stronger connection between the reduction relation and the sets of approximate normal forms holds, i.e.:

$$M =_a N \Rightarrow \mathcal{A}(M) = \mathcal{A}(N).$$

This will follow from the subject conversion of $\mathcal{L}$ (Theorem 49) and the full abstraction of the filter model (Theorem 83).

Now we can prove for our calculus a standard property of λ-calculus: a term is solvable iff it has an approximant different from Ω.

Proposition 27.

(i) $M \in \mathsf{SOL} \iff \mathcal{C}(M) \neq \emptyset$;
(ii) $M \in \mathsf{SOL} \iff \mathcal{A}(M) \neq \{\Omega\}$.

Proof. *(i)* follows from Definitions 17, 19*(ii)* and Corollary 9.
(ii) is a consequence of *(i)* and of Definition 24. □

One would expect $\sqsubseteq^{\mathcal{A}}$ to be a refinement of $\sqsubseteq^{\mathcal{O}}$; this is in fact true. A direct proof based on an approximation theorem á la Wadsworth [45] is possible, but we will obtain it for free from the adequacy and full abstraction results of Section 6.

4 Simple Types and Semimodels

In this section we type the terms of our calculus by means of simple types and we define a set semimodel in the sense of [39].

4.1 The Type Assignment System $\mathcal{B}$

Curry types are thought of as properties of terms. The properties in which we are mainly interested concern solvability. This guides the choice of typing rules for $+$ and $||$.
Indeed to assure that $M + N$ normalizes with respect to $\longrightarrow^h$, we have to prove that both M and N have the same property. Generalizing to arbitrary properties we type $M + N$ with σ if both M and N can be typed with σ. This is also the choice of [1].
Conversely, for $M||N$ to be normalizable it suffices that either M or N normalizes. Extending this notion to arbitrary properties, it follows that one is entitled to type $M||N$ with σ as soon as M or N (or both) can be typed with σ. See [16] for further motivations.

Let the set *Type* of types be defined by

$$\sigma ::= t \mid \sigma \to \sigma,$$

where t ranges over a denumerable collection of type variables. A *statement* is an expression of the form $M{:}\sigma$, where M is a λ-term and σ a type. A *basis* Γ is a set of statements such that subjects are pairwise distinct variables. $FV(\Gamma)$ is the set of term variables in Γ.

Definition 28 (The System $\mathcal{B}$). The axioms and rules of the basic assignment system $\mathcal{B}$ are the following:

$$(\mathrm{Ax}) \quad \Gamma, x{:}\sigma \vdash x{:}\sigma$$

$$(\to \mathrm{I}) \ \frac{\Gamma, x{:}\sigma \vdash M{:}\tau}{\Gamma \vdash \lambda x.M{:}\sigma \to \tau} \qquad (\to \mathrm{E}) \ \frac{\Gamma \vdash M{:}\sigma \to \tau \quad \Gamma \vdash N{:}\sigma}{\Gamma \vdash MN{:}\tau}$$

$$(+\,\mathrm{I}) \ \frac{\Gamma \vdash M{:}\sigma \quad \Gamma \vdash N{:}\sigma}{\Gamma \vdash M + N{:}\sigma} \qquad (||\,\mathrm{I}) \ \frac{\Gamma \vdash M{:}\sigma}{\Gamma \vdash M||N{:}\sigma} \qquad \frac{\Gamma \vdash N{:}\sigma}{\Gamma \vdash M||N{:}\sigma}$$

If $\Gamma \vdash M{:}\sigma$ is provable in $\mathcal{B}$, we write $\Gamma \vdash_{\mathcal{B}} M{:}\sigma$.

In this system, as in Curry's original one, there is a correspondence between the main constructor of the subject of the conclusion in each rule and the rule itself; this does not hold for the type. However, classical terms (i.e. those without occurrences of $+$ and $||$) have just their simple types. This property results in a simple theory of the type assignment system.

A routine induction on derivations in $\mathcal{B}$ shows:

Lemma 29 (Structural Properties of Deductions in $\mathcal{B}$).

(i) $\Gamma \vdash_{\mathcal{B}} x{:}\tau \;\Leftrightarrow\; x{:}\tau \in \Gamma$;
(ii) $\Gamma \vdash_{\mathcal{B}} \lambda x.M{:}\sigma \to \tau \;\Leftrightarrow\; \Gamma, x{:}\sigma \vdash_{\mathcal{B}} M{:}\tau$;
(iii) $\Gamma \vdash_{\mathcal{B}} MN{:}\tau \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\sigma \to \tau$ *and* $\Gamma \vdash_{\mathcal{B}} N{:}\sigma$ *for some* σ;
(iv) $\Gamma \vdash_{\mathcal{B}} M + N{:}\sigma \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\sigma$ *and* $\Gamma \vdash_{\mathcal{B}} N{:}\sigma$;
(v) $\Gamma \vdash_{\mathcal{B}} M||N{:}\sigma \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{B}} M{:}\sigma$ *or* $\Gamma \vdash_{\mathcal{B}} N{:}\sigma$.

Using this lemma it is easy to prove the following corollary by induction on the definition of $\longrightarrow_a$. We consider this reduction, since it includes $\longrightarrow$ (which includes $\longrightarrow^h$).

Corollary 30 (Subject Reduction of $\mathcal{B}$). $\Gamma \vdash_{\mathcal{B}} M{:}\sigma$ *and* $M \xrightarrow{*}_a N \Rightarrow \Gamma \vdash_{\mathcal{B}} N{:}\sigma$.

As an immediate consequence of 30, we have the subject reduction property of $\mathcal{B}$ for $\xrightarrow{*}$.

Remark 31. As stated in [19], also $\xrightarrow{*}_{pn}$ enjoys the subject reduction property.

4.2 The Set Semimodel

For the classical λ-calculus, a filter model construction with simple types, even considering as a "filter" any set of types, does not yield a λ-model (see e.g. [25]). Indeed the best one can obtain is a *semimodel* in the sense of [39]. I.e. a model in which interreducible terms are equal, but in general convertible terms are not (M, N are interreducible iff $M \xrightarrow{*} N$ and $N \xrightarrow{*} M$). Adapting Plotkin's definition to the present context (see also [1]) we introduce the following notion:

Definition 32. A *semimodel* is a structure

$$\mathcal{P} = \langle P, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{P}} \rangle$$

where $\langle P, \sqsubseteq \rangle$ is a poset, and $\cdot, \sqcap, \sqcup$ are binary monotonic operations that satisfy the following requirements:

$$d \sqcap e \sqsubseteq d, d \sqcap e \sqsubseteq e, d \sqsubseteq d \sqcup e, e \sqsubseteq d \sqcup e$$
$$\text{and} \qquad (d \sqcup d') \cdot e \sqsubseteq (d \cdot e) \sqcup (d' \cdot e).$$

Finally $[\![\cdot]\!]^{\mathcal{P}}: \Lambda_{+\|} \times Env \to P$, where $Env = \{\rho \mid \rho: TermVar \to P\}$, is such that:

(a) $[\![M + N]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \sqcap [\![N]\!]^{\mathcal{P}}_\rho$;
(b) $[\![M \| N]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \sqcup [\![N]\!]^{\mathcal{P}}_\rho$;
(c) $[\![x]\!]^{\mathcal{P}}_\rho = \rho(x)$;
(d) $[\![MN]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_\rho \cdot [\![N]\!]^{\mathcal{P}}_\rho$;
(e) $\forall d \in P.\ [\![\lambda x.M]\!]^{\mathcal{P}}_\rho \cdot d \sqsubseteq [\![M]\!]^{\mathcal{P}}_{\rho[d/x]}$;
(f) $\forall x \in \mathrm{FV}(M).\ \rho(x) = \rho'(x) \Rightarrow [\![M]\!]^{\mathcal{P}}_\rho = [\![M]\!]^{\mathcal{P}}_{\rho'}$;
(g) $(\forall d \in P.\ [\![M]\!]^{\mathcal{P}}_{\rho[d/x]} \sqsubseteq [\![N]\!]^{\mathcal{P}}_{\rho[d/x]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{P}}_\rho \sqsubseteq [\![\lambda x.N]\!]^{\mathcal{P}}_\rho$.

Semimodels interpret the reduction relation, as stated in the following proposition, which can be proved by induction on the definition of $\longrightarrow_a$. In the case of $(M + N)L \longrightarrow_a ML + NL$ this follows from the monotonicity of the application which implies $(d \sqcap d') \cdot e \sqsubseteq (d \cdot e) \sqcap (d' \cdot e)$.

Proposition 33. *$M \xrightarrow{*}_a N \Rightarrow \forall \rho.\ [\![M]\!]^{\mathcal{P}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{P}}_\rho$ for all semimodels $\mathcal{P}$.*

Notice that Proposition 33 holds even if $\xrightarrow{*}_a$ is replaced by $\xrightarrow{*}_{pn}$. In the case of the classical λ-calculus one has $\Leftrightarrow$ (see [39]). Here, instead, completeness with respect to reduction does not hold: e.g. we have, by definition, that $\forall \rho \in Env.\ [\![M]\!]_\rho \sqsubseteq [\![M \| N]\!]_\rho$ but we do not have $M \xrightarrow{*}_a M \| N$. This does not seem to be unfortunate; indeed we are looking for a partial order (and its relative equivalence) which is, in a sense, more abstract than reducibility.

As expected, the type assignment $\mathcal{B}$ induces a semimodel.

Proposition 34. *For $a, b \subseteq Type$, let $a \cdot b = \{\tau \in Type \mid \exists \sigma \in b.\ \sigma \to \tau \in a\}$ and*

$$[\![M]\!]^{\mathcal{B}}_\rho = \{\sigma \mid \Gamma \vdash_{\mathcal{B}} M{:}\sigma \text{ for some } \Gamma \subseteq \{x{:}\tau \mid \tau \in \rho(x)\}\}.$$

The structure

$$\langle \wp(Type), \subseteq, \cdot, \cap, \cup, [\![\cdot]\!]^{\mathcal{B}} \rangle$$

is a semimodel (the set semimodel*).*

The interpretation of the parallel and non-deterministic constructors in the set semimodel can also be easily stated using set theoretic operators. I.e., for all ρ:

$$[\![M+N]\!]^{\mathcal{B}}_{\rho} = [\![M]\!]^{\mathcal{B}}_{\rho} \cap [\![N]\!]^{\mathcal{B}}_{\rho} \text{ and } [\![M||N]\!]^{\mathcal{B}}_{\rho} = [\![M]\!]^{\mathcal{B}}_{\rho} \cup [\![N]\!]^{\mathcal{B}}_{\rho}.$$

To interpret types over a given semimodel we use the *simple semantics* of types (see [23, 39]).

Definition 35. A *type structure* over $\mathcal{P} = \langle P, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{P}} \rangle$ is a pair $\langle \mathcal{T}, \Rightarrow \rangle$ where:

(i) $\mathcal{T} \subseteq \{X \in \wp(P) \mid X$ is not empty, upper closed and $d, e \in X$ imply $d \sqcap e \in X\}$;

(ii) $\Rightarrow$ is a binary function over $\mathcal{T}$ such that

(a) $X \Rightarrow Y \subseteq \{d \in P \mid \forall e \in X.\, d \cdot e \in Y\}$,
(b) $d \in X$ and $[\![M]\!]^{\mathcal{P}}_{\rho[d/x]} \in Y$ imply $[\![\lambda x.M]\!]^{\mathcal{P}}_{\rho} \in X \Rightarrow Y$,
for all $X, Y \in \mathcal{T}$.

Definition 36.

(i) A *type environment* is a map η from type variables to $\mathcal{T}$.

(ii) $[\![\sigma]\!]^{\mathcal{T}}_{\eta} \in \mathcal{T}$ is defined by

$$[\![t]\!]^{\mathcal{T}}_{\eta} = \eta(t) \quad \text{and} \quad [\![\sigma \to \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \Rightarrow [\![\tau]\!]^{\mathcal{T}}_{\eta}.$$

(iii) A basis Γ *satisfies* ρ and η iff, for all $x{:}\tau \in \Gamma$, $\rho(x) \in [\![\tau]\!]^{\mathcal{T}}_{\eta}$.

(iv) $\Gamma \models M{:}\sigma \;\Leftrightarrow\; \forall \mathcal{P}, \langle \mathcal{T}, \Rightarrow \rangle$ over $\mathcal{P}$, ρ, η. Γ satisfies $\rho, \eta \Rightarrow [\![M]\!]^{\mathcal{P}}_{\rho} \in [\![\sigma]\!]^{\mathcal{T}}_{\eta}$.

Theorem 37 (Completeness of $\mathcal{B}$). $\Gamma \vdash_{\mathcal{B}} M{:}\sigma \;\Leftrightarrow\; \Gamma \models M{:}\sigma$.

Proof. This proof essentially adapts Plotkin's completeness proof in [39].
($\Rightarrow$) Simple induction on the derivation of $\Gamma \vdash M{:}\sigma$. If the last applied rule is ($\to$ I), the thesis follows from 35*(ii) (b)*. For rule (+ I) use 35*(i)*.
($\Leftarrow$) Using the set semimodel. If we define:

$$\chi_{\sigma} = \{a \subseteq \mathit{Type} \mid \sigma \in a\}, \mathcal{T} = \{\chi_{\sigma}\}_{\sigma \in \mathit{Type}}, \text{ and } \chi_{\sigma} \Rightarrow \chi_{\tau} = \chi_{\sigma \to \tau},$$

then the pair $\langle \mathcal{T}, \Rightarrow \rangle$ is a type structure for the set semimodel.
We take ρ and η such that $\rho(x) = \{\sigma \mid x{:}\sigma \in \Gamma\}$ for every term variable x and $\eta(t) = \chi_t$ for every type variable t. Then we have $[\![\sigma]\!]^{\mathcal{T}}_{\eta} = \chi_{\sigma}$ for all $\sigma \in \mathit{Type}$ and $[\![M]\!]^{\mathcal{B}}_{\rho} \in [\![\sigma]\!]^{\mathcal{T}}_{\eta}$, which imply $\Gamma \vdash_{\mathcal{B}} M{:}\sigma$. □

The set semimodel allows to define a preorder over terms which is a precongruence:

$$M \sqsubseteq^{\mathcal{B}} N \Leftrightarrow_{def} \forall \rho.\; [\![M]\!]^{\mathcal{B}}_{\rho} \subseteq [\![N]\!]^{\mathcal{B}}_{\rho}.$$

We list in the following proposition the main (in)-equations holding in the set semimodel semantics.

Proposition 38. *Let* $\simeq^{\mathcal{B}} = \sqsubseteq^{\mathcal{B}} \cap \sqsupseteq^{\mathcal{B}}$, *then:*

(i) $(\lambda x.M)N \sqsubseteq^{\mathcal{B}} M[N/x]$;
(ii) $(M+N)L \sqsubseteq^{\mathcal{B}} ML + NL$;
(iii) $L(M+N) \sqsubseteq^{\mathcal{B}} LM + LN$;
(iv) $(M||N)L \simeq^{\mathcal{B}} ML||NL$;
(v) $L(M||N) \simeq^{\mathcal{B}} LM||LN$;
(vi) $\lambda x.(M+N) \simeq^{\mathcal{B}} \lambda x.M + \lambda x.N$;
(vii) $\lambda x.(M||N) \simeq^{\mathcal{B}} \lambda x.M||\lambda x.N$;
(viii) $M+N \sqsubseteq^{\mathcal{B}} M, N$;
(ix) $L \sqsubseteq^{\mathcal{B}} M, N \Rightarrow L \sqsubseteq^{\mathcal{B}} M+N$;
(x) $M, N \sqsubseteq^{\mathcal{B}} M||N$;
(xi) $M, N \sqsubseteq^{\mathcal{B}} L \Rightarrow M||N \sqsubseteq^{\mathcal{B}} L$,

where the inequalities (i), (ii) and (iii) are in general proper.

Proof. By the Completeness of $\mathcal{B}$ (Theorem 37) we have

$$M \sqsubseteq^{\mathcal{B}} N \Leftrightarrow \forall \Gamma, \sigma.\ \Gamma \vdash_{\mathcal{B}} M{:}\sigma \Rightarrow \Gamma \vdash_{\mathcal{B}} N{:}\sigma.$$

The positive statements are straightforward consequences of the structural properties of deductions (Lemma 29). To prove that the inequality *(i)* is proper observe that *(i)* essentially claims that the set semimodel is not a λ-model. To see *(ii)*, let

$$\Gamma = \{x{:}\sigma_1 \to \tau, y{:}\sigma_2 \to \tau, z{:}\sigma_1, v{:}\sigma_2\}$$

where $\sigma_1 \not\equiv \sigma_2$. Then $\Gamma \vdash_{\mathcal{B}} x(z\|v) + y(z\|v){:}\,\tau$, but $\Gamma \not\vdash_{\mathcal{B}} (x+y)(z\|v){:}\,\tau$ since $x+y$ has no type. Similarly, for *(iii)*, we have that $\Gamma \vdash_{\mathcal{B}} (x\|y)z + (x\|y)v{:}\,\tau$, but $\Gamma \not\vdash_{\mathcal{B}} (x\|y)(z+v){:}\,\tau$ since $z+v$ has no type. □

Comparing the properties of $\sqsubseteq^{\mathcal{B}}$ with those of $\sqsubseteq^{\mathcal{O}}$ (Proposition 15) and of $\sqsubseteq^{\mathcal{A}}$ (Proposition 60) it turns out that the set semimodel does not agree neither with the operational semantics á la Morris nor with the inclusion of sets of approximants. This failure suggests us to look at a more expressive type assignment system.

5 Intersection, Union Types and λ-lattices

In this section we extend the notion of filter model introduced in [13] to our calculus, the aim being this time to interpret the terms of $\Lambda_{+\|}$ in such a way that the usual λ-calculus equations hold and which fits better the operational behaviour of $+$ and $\|$.

5.1 The Set of Types and its Preorder

Let us redefine the syntax of types as follows:

$$\sigma ::= t \mid \omega \mid \sigma \to \sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma,$$

and call again *Type* the resulting set. In writing types, we assume that $\wedge$ and $\vee$ take precedence over $\to$.

It is clear that to build a filter model a critical choice is that of the preorder between types, since this preorder will appear in a subtyping rule.

Definition 39.

(i) Let $\leq$ be the smallest preorder over types s.t. $\langle Type, \leq\rangle$ is a distributive lattice, in which $\wedge$ is the meet, $\vee$ is the join and ω is the top, and moreover the arrow satisfies:

- *(a)* $\omega \leq \omega \to \omega$;
- *(b)* $(\sigma \to \mu) \wedge (\sigma \to \tau) \leq \sigma \to \mu \wedge \tau$;
- *(c)* $\sigma' \leq \sigma, \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$.

(ii) Let $\sigma = \tau$ mean $\sigma \leq \tau \leq \sigma$.

The subtype relation $\leq$ can be presented axiomatically by adding the inequalities (a)-(c) to any standard axiomatization of distributive lattices. For proof purpose we assume that such a presentation has been fixed.

We need some properties of the $\leq$ relation, whose proof requires a stratification of *Type*.

Definition 40 (Stratification of *Type*). Let us define three subsets T_0, T_1, T_2 of *Type* recursively:

- $t \in T_0$;
- $\omega \in T_2$;
- $\sigma \in T_2, \tau \in T_1 \Rightarrow \sigma \to \tau \in T_0$;
- $n \geq 1, \sigma_1, \ldots, \sigma_n \in T_0 \Rightarrow \sigma_1 \vee \ldots \vee \sigma_n \in T_1$;
- $n \geq 1, \sigma_1, \ldots, \sigma_n \in T_1 \Rightarrow \sigma_1 \wedge \ldots \wedge \sigma_n \in T_2$.

Remark 41. Notice that the set T_2, when restricted to types without $\vee$ occurrences, coincides with the set of normal type schemes of [24] and the set of strict types of [8]. Normal type schemes in [24] were introduced to prove the properties stated in Lemma 45 (for types without $\vee$). Strict types, instead, have been introduced with a different preorder to obtain a syntax directed type assignment system in [8, 10].

Taking $n = 1$ in the clauses above, one sees that $T_0 \subseteq T_1 \subseteq T_2$, and such inclusions are clearly proper.

Over each of these sets we introduce a preorder.

Definition 42. $\leq_i \subseteq T_i \times T_i$ is the least preorder such that:

$(\leq_0)$ $\sigma \leq_0 \tau \;\Leftrightarrow\; \sigma \equiv \tau$ or ($\sigma \equiv \sigma' \to \sigma''$ and $\tau \equiv \tau' \to \tau''$ and $\tau' \leq_2 \sigma'$ and $\sigma'' \leq_1 \tau''$);

$(\leq_1)$ $\sigma_1 \vee \ldots \vee \sigma_n \leq_1 \tau_1 \vee \ldots \vee \tau_m \;\Leftrightarrow\; \forall i \leq n\, \exists j \leq m.\ \sigma_i \leq_0 \tau_j$;

$(\leq_2)$ $\sigma \leq_2 \tau \;\Leftrightarrow\; \tau \equiv \omega$ or ($\sigma \equiv \sigma_1 \wedge \ldots \wedge \sigma_n$, $\tau \equiv \tau_1 \wedge \ldots \wedge \tau_m$ and $\forall j \leq m\, \exists i \leq n.\ \sigma_i \leq_1 \tau_j$).

For each type in *Type* we can find an equivalent type in T_2; this means that we can limit ourself to consider types in T_2, provided that there is a map $(\ )^*$ associating to each type in *Type* a standard form in T_2.

Notation. In writing $\tau^* \equiv \bigwedge_{i \in I} \tau_i$ we assume that $\tau_i \in T_1$ for all $i \in I$.

Definition 43. The map $(\ )^*: Type \to T_2$ is defined by:

$t^* \equiv t,\ \omega^* \equiv \omega$

$$(\sigma \to \tau)^* \equiv \begin{cases} \bigwedge_{i \in I}(\sigma^* \to \tau_i) & \text{if } \tau^* \equiv \bigwedge_{i \in I} \tau_i \text{ and } \tau^* \not\equiv \omega \\ \omega & \text{otherwise} \end{cases}$$

$$(\sigma \vee \tau)^* \equiv \begin{cases} \bigwedge_{i \in I} \bigwedge_{j \in J}(\sigma_i \vee \tau_j) & \text{if } \sigma^* \equiv \bigwedge_{i \in I} \sigma_i,\ \sigma^* \not\equiv \omega \text{ and } \tau^* \equiv \bigwedge_{j \in J} \tau_j,\ \tau^* \not\equiv \omega \\ \omega & \text{otherwise} \end{cases}$$

$$(\sigma \wedge \tau)^* \equiv \begin{cases} \sigma^* & \text{if } \tau^* \equiv \omega \\ \tau^* & \text{if } \sigma^* \equiv \omega \\ \sigma^* \wedge \tau^* & \text{otherwise.} \end{cases}$$

Proposition 44. *For all $\sigma, \tau \in Type$:*

(i) $\sigma = \sigma^*$;

(ii) $\sigma, \tau \in T_i, \sigma \leq_i \tau \Rightarrow \sigma \leq \tau$ *for* $i = 0, 1, 2$;

(iii) $\sigma \leq \tau \Rightarrow \sigma^* \leq_2 \tau^*$.

Proof. *(i)* By induction on the definition of the map $(\ )^*$.

(ii) By induction on the definition of $\leq_i$.

(iii) By *(i)* it suffices to show that $\sigma^* \leq \tau^*$ implies $\sigma^* \leq_2 \tau^*$. This is done by induction on the formal derivation of $\sigma^* \leq \tau^*$. □

Lemma 45.

(i) $\mu \wedge \nu \leq \sigma \to \tau$ *and* $\mu \neq \omega$ *and* $\nu \neq \omega \Rightarrow \exists \tau_1, \tau_2.\ \tau = \tau_1 \wedge \tau_2$ *and* $\mu \leq \sigma \to \tau_1$ *and* $\nu \leq \sigma \to \tau_2$;

(ii) $\bigwedge_{i\in I}(\mu_i \to \nu_i) \leq \sigma \to \tau$ *and* $\tau \neq \omega \Rightarrow \exists J \subseteq I.\ \sigma \leq \bigwedge_{j\in J}\mu_j$ *and* $\bigwedge_{j\in J}\nu_j \leq \tau$.

Proof. *(i)* : let

$$(\mu \wedge \nu)^* = \bigwedge_{i\in I}\mu_i \wedge \bigwedge_{j\in J}\nu_j \text{ and } (\sigma \to \tau)^* = \bigwedge_{k\in K}(\sigma^* \to \pi_k),$$

supposing $\mu^* = \bigwedge_{i\in I}\mu_i, \nu^* = \bigwedge_{j\in J}\nu_j$ and $\tau^* = \bigwedge_{k\in K}\pi_k$. Using 44*(i)*, *(ii)*, *(iii)* and the definition of $\leq_2$, we have that

$$\forall k.\ (\exists i.\ \mu_i \leq_1 \sigma^* \to \pi_k) \text{ or } (\exists j.\ \nu_i \leq_1 \sigma^* \to \pi_k).$$

Therefore we can choose τ_1 as the intersection of the π_k which satisfy the first inequality and τ_2 as the intersection of the remaining π_k. If one of these intersections is empty, we choose ω for the corresponding τ_i $(i = 1, 2)$.

(ii) : let $\nu_i^* = \bigwedge_{l\in L}\nu_{i,l}$ (where L depends on i) and $\tau^* = \bigwedge_{k\in K}\tau_k$. Then, by 44 *(iii)* and Definition 43,

$$\bigwedge_{i\in I}(\mu_i \to \nu_i) \leq \sigma \to \tau \Rightarrow \bigwedge_{i\in I}\bigwedge_{l\in L}(\mu_i^* \to \nu_{i,l}) \leq_2 \bigwedge_{k\in K}(\sigma^* \to \tau_k).$$

It follows that

$$\forall k\ \exists i, l.\ \mu_i^* \to \nu_{i,l} \leq_1 \sigma^* \to \tau_k,$$

which in this case is equivalent to

$$\forall k\ \exists i, l.\ \mu_i^* \to \nu_{i,l} \leq_0 \sigma^* \to \tau_k,$$

and hence

$$\forall k\ \exists i, l.\ \sigma^* \leq_2 \mu_i^* \text{ and } \nu_{i,l} \leq_1 \tau_k.$$

So we conclude

$$\forall k\ \exists i.\ \sigma \leq \mu_i \text{ and } \bigwedge_{l\in L}\nu_{i,l} \leq_2 \tau_k.$$

Taking J as the set of i's which satisfy these inequalities for some $k \in K$, we are done. □

5.2 The Type Assignment System $\mathcal{L}$

We introduce now a type assignment system for our extended language of types. We add a rule (ω) which takes into account the universal character of ω, and two standard rules of introduction of $\wedge$ and $\vee$. Moreover we use the preorder on types defined in previous section in a subtyping rule.

Notice that a rule of $\wedge$ elimination is derivable, while a rule of $\vee$ elimination would be unsound (see Remark 47*(ii)*).

Definition 46. The system $\mathcal{L}$ is obtained by adding to the basic system $\mathcal{B}$ the following axiom and rules:

$$(\omega)\quad \Gamma \vdash M:\omega \qquad (\wedge\mathrm{I})\ \frac{\Gamma \vdash M:\sigma \quad \Gamma \vdash M:\tau}{\Gamma \vdash M:\sigma \wedge \tau}$$

$$(\vee\mathrm{I})\ \frac{\Gamma \vdash M:\sigma}{\Gamma \vdash M:\sigma \vee \tau} \qquad \frac{\Gamma \vdash M:\tau}{\Gamma \vdash M:\sigma \vee \tau} \quad (\leq)\ \frac{\Gamma \vdash M:\sigma \quad \sigma \leq \tau}{\Gamma \vdash M:\tau}$$

If $\Gamma \vdash M:\sigma$ is provable in the system $\mathcal{L}$ we write $\Gamma \vdash_{\mathcal{L}} M:\sigma$.

Notation. In the following we shall sometimes refer to the stronger basis which can be formed out of two given bases. This is done by taking the intersection of the types which are predicates of the same variable:

$$\begin{aligned}\Gamma \uplus \Gamma' = & \{x:\sigma \wedge \tau \mid x:\sigma \in \Gamma \text{ and } x:\tau \in \Gamma'\} \\ & \cup \{x:\sigma \mid x:\sigma \in \Gamma \text{ and } x \notin FV(\Gamma')\} \\ & \cup \{x:\tau \mid x:\tau \in \Gamma' \text{ and } x \notin FV(\Gamma)\}.\end{aligned}$$

Accordingly we define:

$$\Gamma \Subset \Gamma' \ \Leftrightarrow\ \exists \Gamma''.\, \Gamma \uplus \Gamma'' = \Gamma'.$$

Remark 47. *(i)* Of course rule $(\vee\mathrm{I})$ is derivable. The following rules are admissible:

$$\frac{\Gamma \vdash M:\sigma \quad \Gamma \vdash N:\tau}{\Gamma \vdash M+N:\sigma \vee \tau} \qquad \frac{\Gamma \vdash M:\sigma \quad \Gamma \vdash N:\tau}{\Gamma \vdash M||N:\sigma \wedge \tau}$$

$$\frac{\Gamma \vdash M:\sigma \wedge \tau}{\Gamma \vdash M:\sigma} \qquad \frac{\Gamma \vdash M:\sigma \wedge \tau}{\Gamma \vdash M:\tau} \qquad \frac{\Gamma, x:\sigma \vdash M:\tau \quad \sigma' \leq \sigma}{\Gamma, x:\sigma' \vdash M:\tau}\,.$$

(ii) A natural rule of $\vee$ elimination in the present setting would be:

$$(\vee\mathrm{E})\ \frac{\Gamma, x:\sigma \vdash M:\mu \qquad \Gamma, x:\tau \vdash M:\mu \qquad \Gamma \vdash N:\sigma \vee \tau}{\Gamma \vdash M[N/x]:\mu}.$$

This is a rule of the system proposed in [11], where only pure λ-terms are considered. In presence of + and of the corresponding typing rule, however, rule $(\vee\mathrm{E})$ causes the loss of the subject reduction property (established below in Theorem 49).

Moreover with $(\vee\mathrm{E})$ we would lose also the property (proved in Corollary 69) that unsolvable terms have only types equivalent to ω.

We give an example showing both failures. Let $\mathbf{I}, \mathbf{K}, \Delta$ be as in the proof of Proposition 15, and $\mathbf{O} \equiv \lambda xy.y$, then we have:

$$\begin{aligned} & x:(\nu \to \omega \to \nu) \wedge \nu \vdash_{\mathcal{L}} xx\mathbf{KI}(\Delta\Delta):\mu, \\ & x:\omega \to \nu \to \nu \vdash_{\mathcal{L}} xx\mathbf{KI}(\Delta\Delta):\mu, \\ \text{and} \quad & \vdash_{\mathcal{L}} \mathbf{K}+\mathbf{O}:((\nu \to \omega \to \nu) \wedge \nu) \vee (\omega \to \nu \to \nu), \\ \text{where} \quad & \mu \equiv t \to t,\ \nu \equiv \mu \to \omega \to \mu. \end{aligned}$$

This can be easily checked considering that

$$\vdash_{\mathcal{L}} \mathbf{I}:\mu,\ \ \vdash_{\mathcal{L}} \mathbf{K}:(\nu \to \omega \to \nu) \wedge \nu \text{ and } \vdash_{\mathcal{L}} \mathbf{O}:\omega \to \nu \to \nu.$$

Therefore using $(\vee\mathrm{E})$ we could derive:

$$M \equiv (\mathbf{K}+\mathbf{O})(\mathbf{K}+\mathbf{O})\mathbf{KI}(\Delta\Delta):\mu.$$

But M reduces to $\mathbf{I}+\Delta\Delta+\mathbf{I}+\mathbf{I}$ and therefore it is unsolvable. We lose subject reduction, since only type ω can be deduced for $\Delta\Delta$, and hence for $\mathbf{I}+\Delta\Delta+\mathbf{I}+\mathbf{I}$. Moreover M is unsolvable but it has type $\mu \not\sim \omega$.

(iii) Notice that

$$\sigma \vee \tau \to \mu \le (\sigma \to \mu) \wedge (\tau \to \mu),$$

but the converse does not hold. The equality is derivable in the system proposed in [11]. In the present system, by postulating

$$(\sigma \to \mu) \wedge (\tau \to \mu) \le \sigma \vee \tau \to \mu$$

we would have the same problems we discussed in *(ii)* with rule $(\vee\mathrm{E})$. In fact the following derivation would be possible:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x{:}\sigma \vdash M{:}\mu}{\Gamma \vdash \lambda x.M{:}\sigma \to \mu}(\to \mathrm{I}) \quad \dfrac{\Gamma, x{:}\tau \vdash M{:}\mu}{\Gamma \vdash \lambda x.M{:}\tau \to \mu}(\to \mathrm{I})}{\Gamma \vdash \lambda x.M{:}(\sigma \to \mu) \wedge (\tau \to \mu)}(\wedge \mathrm{I})}{\Gamma \vdash \lambda x.M{:}\sigma \vee \tau \to \mu}(\le) \qquad \Gamma \vdash N{:}\sigma \vee \tau}{\Gamma \vdash (\lambda x.M)N{:}\mu}(\to \mathrm{E})$$

If we compare this derivation with the $(\vee\mathrm{E})$ rule we see that from the same premises we obtain the same type for a β-expansion of the subject.

(iv) In a λ-calculus enriched with constants (and with the corresponding constant types) in the standard way, the typing rules for $+$ and $||$ give a sort of abstract interpretation [26, 18]. As an example we would have that $1 + \mathtt{true}$ has type *integer* $\vee$ *boolean* and $1||\mathtt{true}$ has both types *integer* and *boolean*.

For the present type assignment system the proof of structural properties is a bit more involved than in case of system $\mathcal{B}$. If $x : \sigma \in \Gamma$, then we define $\Gamma(x) =_{Df} \sigma$.

Lemma 48 (Structural Properties of Deductions in $\mathcal{L}$).

(i) *If* $\tau \neq \omega$, *then* $\Gamma \vdash_{\mathcal{L}} x{:}\tau \;\Leftrightarrow\; \Gamma(x) \le \tau$;

(ii) $\Gamma \vdash_{\mathcal{L}} \lambda x.M{:}\tau \;\Leftrightarrow$
$\exists\, \mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n.\ \Gamma \vdash_{\mathcal{L}} \lambda x.M{:}\ \bigwedge_{i=1}^{n}(\mu_i \to \nu_i)$ *and* $\bigwedge_{i=1}^{n}(\mu_i \to \nu_i) \le \tau$;

(iii) $\Gamma \vdash_{\mathcal{L}} \lambda x.M{:}\sigma \to \tau \;\Leftrightarrow\; \Gamma, x{:}\sigma \vdash_{\mathcal{L}} M{:}\tau$;

(iv) $\Gamma \vdash_{\mathcal{L}} MN{:}\tau \;\Leftrightarrow\; \exists \sigma.\ \Gamma \vdash_{\mathcal{L}} M{:}\sigma \to \tau$ *and* $\Gamma \vdash_{\mathcal{L}} N{:}\sigma$;

(v) $\Gamma \vdash_{\mathcal{L}} M + N{:}\sigma \;\Leftrightarrow\; \Gamma \vdash_{\mathcal{L}} M{:}\sigma$ *and* $\Gamma \vdash_{\mathcal{L}} N{:}\sigma$;

(vi) $\Gamma \vdash_{\mathcal{L}} M||N{:}\tau \;\Leftrightarrow\; \exists \sigma, \sigma'.\ \sigma \wedge \sigma' \le \tau$ *and* $\Gamma \vdash_{\mathcal{L}} M{:}\sigma$ *and* $\Gamma \vdash_{\mathcal{L}} N{:}\sigma'$.

Proof. *(i)* and *(iv)*: it is easy to extend to union types the proof given in [13].

In *(ii)*, *(iii)* *(v)* and *(vi)*, $\Leftarrow$ is immediate. We show $\Rightarrow$.

(ii) If $\tau = \omega$ we can take $n = 1$, $\mu_1 = \nu_1 = \omega$, since $\Gamma \vdash \lambda x.M{:}\omega \to \omega$ is provable in $\mathcal{L}$. Otherwise choose a derivation of $\Gamma \vdash \lambda x.M{:}\tau$. Being $\tau \neq \omega$ rule $(\to \mathrm{I})$ has been used. Let

$$\Gamma \vdash \lambda x.M{:}\mu_1 \to \nu_1, \ldots, \Gamma \vdash \lambda x.M{:}\mu_n \to \nu_n$$

be the conclusions of all $(\to \mathrm{I})$ rules having $\lambda x.M$ as subject in this derivation. Now $\lambda x.M$ is the same subject of the conclusion of the derivation itself; hence below such rules only $(\le)$ and $(\wedge\ \mathrm{I})$ rules are possible. This implies that

$$(\mu_1 \to \nu_1) \wedge \cdots \wedge (\mu_n \to \nu_n) \le \tau.$$

(iii) Assume $\tau \neq \omega$. Let $\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n$ be as in the proof of *(ii)*. Then by *(ii)* itself:

$$(\mu_1 \to \nu_1) \wedge \cdots \wedge (\mu_n \to \nu_n) \leq \sigma \to \tau$$

so that, by Lemma 45 *(ii)*,

$$\exists J \subseteq \{1, \cdots, n\}. \, \sigma \leq \bigwedge_{j \in J} \mu_j \text{ and } \bigwedge_{j \in J} \nu_j \leq \tau.$$

On the other hand the premises of the $(\to \mathrm{I})$ rules are of the shape $\Gamma, x{:}\mu_i \vdash M{:}\nu_i$ and have been derived for $1 \leq i \leq n$. Hence $\Gamma, x{:}\sigma \vdash_{\mathcal{L}} M{:}\tau$.

(v) Let a deduction of $\Gamma \vdash M + N{:}\sigma$ be given and let

$$\Gamma \vdash M + N{:}\sigma_1, \ldots, \Gamma \vdash M + N{:}\sigma_n$$

be all the statements in this deduction on which $\Gamma \vdash M + N{:}\sigma$ depends and which are conclusions of rule $(+\mathrm{I})$. Then

$$\sigma_1 \wedge \cdots \wedge \sigma_n \leq \sigma \text{ and } \Gamma \vdash_{\mathcal{L}} M{:}\sigma_i, \Gamma \vdash_{\mathcal{L}} N{:}\sigma_i,$$

for $1 \leq i \leq n$. So we can derive $\Gamma \vdash M{:}\sigma$ and $\Gamma \vdash N{:}\sigma$ using $(\wedge \mathrm{I})$ and $(\leq)$.

(vi) Finally, given a deduction of $\Gamma \vdash M||N{:}\tau$, let

$$\Gamma \vdash M||N{:}\sigma_1, \ldots, \Gamma \vdash M||N{:}\sigma_n$$

be all the statements in this deduction on which $\Gamma \vdash M||N{:}\tau$ depends and which are conclusions of rule $(||\mathrm{I})$. Then

$$\sigma_1 \wedge \cdots \wedge \sigma_n \leq \tau \text{ and } \forall\, i \leq n.\ (\Gamma \vdash_{\mathcal{L}} M{:}\sigma_i \text{ or } \Gamma \vdash_{\mathcal{L}} N{:}\sigma_i).$$

We assume, without loss of generality, that, for some h, $\Gamma \vdash_{\mathcal{L}} M{:}\sigma_i$ for $1 \leq i \leq h$ and $\Gamma \vdash_{\mathcal{L}} N{:}\sigma_j$ for $h+1 \leq j \leq n$. It follows that, by rule $(\wedge \mathrm{I})$, $\Gamma \vdash M{:}\sigma$ and $\Gamma \vdash N{:}\sigma'$ are provable, where $\sigma \equiv \sigma_1 \wedge \ldots \wedge \sigma_h$, $\sigma' \equiv \sigma_{h+1} \wedge \ldots \wedge \sigma_n$ and $\sigma \wedge \sigma' \leq \tau$. □

The invariance of types under subject conversion with respect to $=_a$ is now an easy consequence of the previous Lemmas. We consider $=_a$, since it includes $=$.

Theorem 49 (Subject Conversion of $\mathcal{L}$).

$$\Gamma \vdash_{\mathcal{L}} M{:}\sigma \textit{ and } M =_a N \Rightarrow \Gamma \vdash_{\mathcal{L}} N{:}\sigma.$$

Proof. It sufficies to prove the thesis when $M =_a N$ is replaced by $M \xrightarrow{*}_a N$ (subject reduction) and by $N \xrightarrow{*}_a M$ (subject expansion). We show this by induction on the definition of $\longrightarrow_a$.

The most interesting case is $(P||Q)L \longrightarrow_a PL||QL$. Let $\Gamma \vdash_{\mathcal{L}} (P||Q)L{:}\tau$; then we have, by Lemma 48*(iv)*, that $\Gamma \vdash_{\mathcal{L}} L{:}\sigma$ and $\Gamma \vdash_{\mathcal{L}} P||Q{:}\sigma \to \tau$ for some σ. This implies, by Lemma 48*(vi)*, that there exist μ, ν such that

$$\Gamma \vdash_{\mathcal{L}} P{:}\mu, \Gamma \vdash_{\mathcal{L}} Q{:}\nu \text{ and } \mu \wedge \nu \leq \sigma \to \tau.$$

Assuming $\mu \neq \omega$ and $\nu \neq \omega$ we have, by Lemma 45*(i)*,

$$\exists\, \tau_1, \tau_2.\ \tau = \tau_1 \wedge \tau_2 \text{ and } \mu \leq \sigma \to \tau_1 \text{ and } \nu \leq \sigma \to \tau_2.$$

It follows that $\Gamma \vdash_{\mathcal{L}} P{:}\sigma \to \tau_1$ and $\Gamma \vdash_{\mathcal{L}} Q{:}\sigma \to \tau_2$, so we conclude $\Gamma \vdash_{\mathcal{L}} PL||QL{:}\tau$.
The case in which $\mu = \omega$ or $\nu = \omega$ is similar and simpler.

Viceversa, let $\Gamma \vdash_{\mathcal{L}} PL||QL{:}\,\tau$. By Lemma 48*(vi)* there are μ, ν such that

$$\Gamma \vdash_{\mathcal{L}} PL{:}\,\mu, \Gamma \vdash_{\mathcal{L}} QL{:}\,\nu \text{ and } \mu \wedge \nu \leq \tau.$$

This implies by Lemma 48*(iv)* that there are σ_1, σ_2 such that

$$\Gamma \vdash_{\mathcal{L}} P{:}\,\sigma_1 \to \mu, \Gamma \vdash_{\mathcal{L}} L{:}\,\sigma_1 \text{ and } \Gamma \vdash_{\mathcal{L}} Q{:}\,\sigma_2 \to \nu, \Gamma \vdash_{\mathcal{L}} L{:}\,\sigma_2.$$

Therefore, by rules $(||\,\mathrm{I})$, $(\wedge\,\mathrm{I})$, and $(\leq)$

$$\Gamma \vdash_{\mathcal{L}} P||Q{:}\,\sigma_1 \wedge \sigma_2 \to \mu \wedge \nu \text{ and } \Gamma \vdash_{\mathcal{L}} L{:}\,\sigma_1 \wedge \sigma_2,$$

so that $\Gamma \vdash_{\mathcal{L}} (P||Q)L{:}\,\tau$. □

Remark 50. As an immediate consequence of Theorem 49, we have the subject conversion of $\mathcal{L}$ also for the relation $=$. Instead, as stated in [19], only subject reduction of $\mathcal{L}$ holds for the reduction $\longrightarrow_{pn}$. This is clear looking at rule $(+_{\mathcal{C}})$, because this rule properly increases the set of types of the subject.

5.3 The λ-lattices

As the set semimodel suggests, when interpreting our calculus we naturally get lattices. We make precise now what is a model of this calculus. We do this by incorporating the notion of lattice into that of λ-model of [25].

Definition 51. A *λ-lattice* is a structure $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{D}} \rangle$ where:

(i) $\langle D, \sqsubseteq, \sqcap, \sqcup \rangle$ is a lattice;

(ii) $\cdot : D \times D \to D$ is monotonic;

(iii) $\forall d, d', e \in D.\ (d \sqcup d') \cdot e \sqsubseteq (d \cdot e) \sqcup (d' \cdot e)$ and $(d \cdot e) \sqcap (d' \cdot e) \sqsubseteq (d \sqcap d') \cdot e$;

(iv) $[\![\cdot]\!]^{\mathcal{D}} : Env \times \Lambda_{+||} \to D$, where $Env = \{\rho \mid \rho{:}\ TermVar \to D\}$, is such that:

  (a) $[\![M + N]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho} \sqcap [\![N]\!]^{\mathcal{D}}_{\rho}$;

  (b) $[\![M||N]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho} \sqcup [\![N]\!]^{\mathcal{D}}_{\rho}$;

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

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

  (e) $\forall d \in D.\ [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} \cdot d = [\![M]\!]^{\mathcal{D}}_{\rho[d/x]}$;

  (f) $\forall x \in \mathrm{FV}(M).\ \rho(x) = \rho'(x) \Rightarrow [\![M]\!]^{\mathcal{D}}_{\rho} = [\![M]\!]^{\mathcal{D}}_{\rho'}$;

  (g) $(\forall d \in D.\ [\![M]\!]^{\mathcal{D}}_{\rho[d/x]} = [\![N]\!]^{\mathcal{D}}_{\rho[d/x]}) \Rightarrow [\![\lambda x.M]\!]^{\mathcal{D}}_{\rho} = [\![\lambda x.N]\!]^{\mathcal{D}}_{\rho}$.

Clauses *(iv)* from (c) to (g) define syntactical λ-models (see [25]). They have been written to state explicitly that the map $[\![\cdot]\!]^{\mathcal{D}}$ satisfies these clauses not just on the classical λ-terms, but on the whole set $\Lambda_{+||}$.

It is interesting to relate semimodels and λ-lattices considering the role of the order in the structure. Indeed by Proposition 33 the meaning of a term in a semimodel increases along reduction. In the case of λ-lattices, instead, we have:

Proposition 52. $M =_a N \Rightarrow \forall \rho.\ [\![M]\!]^{\mathcal{D}}_{\rho} = [\![N]\!]^{\mathcal{D}}_{\rho}$ *for all λ-lattices $\mathcal{D}$.*

Proof. By induction on the definition of $\longrightarrow_a$ using the conditions of 51*(iii)*. The proof is a straightforward variant of the analogous proof for classical λ-calculus (see [25] or [12] 5.3.4). □

Moreover it is not difficult to show that we have:

$$M \xrightarrow{*}_{pn} N \Rightarrow \forall\rho.\ [\![M]\!]^{\mathcal{D}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{D}}_\rho$$

for all λ-lattices $\mathcal{D}$, where $\sqsubseteq$ can be proper. Indeed $M + N \xrightarrow{*}_{pn} M$ and in general $[\![M+N]\!]^{\mathcal{D}}_\rho \sqsubset [\![M]\!]^{\mathcal{D}}_\rho$.

As immediate consequence of Proposition 15 we obtain a term model based on the contextual semantics which is a λ-lattice.

Proposition 53. *For $M, N \in \Lambda_{+\|}$ define $[M] = \{M' \in \Lambda_{+\|} \mid M \simeq^{\mathcal{O}} M'\}$, $[M]\cdot[N] = [MN]$, $[M] \sqcup [N] = [M\|N]$, $[M] \sqcap [N] = [M+N]$, and $[M] \sqsubseteq [N]$ iff $M \sqsubseteq^{\mathcal{O}} N$. These definitions induce a λ-lattice, where $[\![M]\!]_\rho = [M[\vec{N}/\vec{x}]]$ when $FV(M) = \vec{x}$ and $\rho(\vec{x}) = [\vec{N}]$.*

The existence of the term model implies an adequacy result.

Corollary 54. $\forall M, N \in \Lambda_{+\|}.\ (\forall\lambda\text{-}lattice\, \mathcal{D}, \forall\rho.\ [\![M]\!]^{\mathcal{D}}_\rho \sqsubseteq [\![N]\!]^{\mathcal{D}}_\rho) \Rightarrow M \sqsubseteq^{\mathcal{O}} N$.

5.4 The Filter λ-lattice

Given the usual notion of *filter*, rules (ω), $(\leq)$ and $(\wedge \mathrm{I})$ imply that, for any Γ and M, $\{\sigma \mid \Gamma \vdash_{\mathcal{L}} M{:}\sigma\}$ is a filter. A filter model construction as in [13] can be carried out. If X is a subset of any poset, then let $\uparrow X$ be its upward closure.

Theorem 55. *Let $\mathcal{F}(Type)$ be the set of filters over Type and define, for $f, f' \in \mathcal{F}(Type)$:*

$$f \overline{\cup} f' = \uparrow\{\sigma \wedge \tau \mid \sigma \in f, \tau \in f'\}, \quad f \cdot f' = \{\tau \mid \exists \sigma \in f'.\ \sigma \to \tau \in f\}.$$

Then $f \overline{\cup} f', f \cdot f' \in \mathcal{F}(Type)$. Moreover the structure

$$\langle \mathcal{F}(Type), \subseteq, \cdot, \cap, \overline{\cup}, [\![\cdot]\!]^{\mathcal{L}} \rangle,$$

where

$$[\![M]\!]^{\mathcal{L}}_\rho = \{\sigma \mid \Gamma \vdash_{\mathcal{L}} M{:}\sigma \text{ for some } \Gamma \subseteq \{x{:}\tau \mid \tau \in \rho(x)\}\},$$

is a λ-lattice (the filter λ-lattice*).*

Proof. $f \overline{\cup} f'$ is the least filter including $f \cup f'$, therefore it is the join wrt inclusion in the set of filters. Since filters are closed under intersection, $\langle \mathcal{F}(Type), \cap, \overline{\cup} \rangle$ is a lattice, so that *(i)* of Definition 51 is satisfied.

It is easy to see that $f \cdot f'$ is a filter too: hence "·" is well defined. Moreover "·" is clearly monotonic in both its arguments. So that also *(ii)* of Definition 51 holds.

Now we prove the first clause of *(iii)*. By definition we know

$$\begin{aligned} \tau \in (f_0 \overline{\cup} f_1) \cdot f_2 &\Rightarrow \exists\, \sigma \in f_2.\ \sigma \to \tau \in f_0 \overline{\cup} f_1 \\ &\Rightarrow \exists\, \sigma \in f_2, \mu \in f_0, \nu \in f_1.\ \mu \wedge \nu \leq \sigma \to \tau. \end{aligned}$$

The more interesting case is $\mu \neq \omega$ and $\nu \neq \omega$. By 45*(i)* there are τ_1, τ_2 such that $\tau = \tau_1 \wedge \tau_2$ and $\mu \leq \sigma \to \tau_1, \nu \leq \sigma \to \tau_2$. Therefore by definition $\tau_1 \in f_0 \cdot f_2$ and $\tau_2 \in f_1 \cdot f_2$, so we can conclude

$\tau \in (f_0 \cdot f_2)\overline{\cup}(f_0 \cdot f_1)$.
The proof of the other clause of *(iii)* is similar and simpler.

Lastly we prove *(iv)*. Lemma 48*(v)* implies that

$$[\![M+N]\!]^{\mathcal{L}}_{\rho} = [\![M]\!]^{\mathcal{L}}_{\rho} \cap [\![N]\!]^{\mathcal{L}}_{\rho}$$

and Lemma 48*(vi)* implies that

$$[\![M||N]\!]^{\mathcal{L}}_{\rho} = [\![M]\!]^{\mathcal{L}}_{\rho} \overline{\cup} [\![N]\!]^{\mathcal{L}}_{\rho}$$

for all ρ. Hence clauses (a) and (b) follow.
The clauses from (c) to (g) follow easily from points $(i), (ii), (iii)$ and *(iv)* of Lemma 48 as in the case of classical λ-calculus. □

Definition 56. Let $\mathcal{D} = \langle D, \sqsubseteq, \cdot, \sqcap, \sqcup, [\![\cdot]\!]^{\mathcal{D}} \rangle$ be a λ-lattice. Then a *type structure* over $\mathcal{D}$ is a pair $\langle \mathcal{T}, \Rightarrow \rangle$ such that $\mathcal{T}$ is a sublattice of the lattice of filters over D, $D \in \mathcal{T}$, and $\Rightarrow$ is a binary function over $\mathcal{T}$ such that $X \Rightarrow Y = \{d \in D \mid \forall e \in X.\ d \cdot e \in Y\}$, for all $X, Y \in \mathcal{T}$. Moreover $\mathcal{T}$ is closed under $\cap$, and $\overline{\cup}$ defined by $X \overline{\cup} Y = \uparrow \{d \sqcap d' \mid d \in X, d' \in Y\}$, where we overload $\overline{\cup}$.
The map $[\![\cdot]\!]^{\mathcal{T}}$, interpreting types over $\mathcal{T}$, is defined as in Definition 36*(iii)*, adding three clauses:
(iii) $[\![\omega]\!]^{\mathcal{T}}_{\eta} = D$;
(iv) $[\![\sigma \wedge \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \cap [\![\tau]\!]^{\mathcal{T}}_{\eta}$;
(v) $[\![\sigma \vee \tau]\!]^{\mathcal{T}}_{\eta} = [\![\sigma]\!]^{\mathcal{T}}_{\eta} \overline{\cup} [\![\tau]\!]^{\mathcal{T}}_{\eta}$.

In the filter λ-lattice defined in Theorem 55, the interpretation of a type turns out to be a filter of filters of types. Since the lattice of types is distributive, the lattice of filters forming the filter λ-lattice is distributive too, hence the upward closure in clause *(v)* above is redundant in this case. The following proposition is proved by routine calculations.

Proposition 57. *Let $\chi_\sigma = \{f \in \mathcal{F}(Type) \mid \sigma \in f\}$. Then $\langle \{\chi_\sigma \mid \sigma \in Type\}, \Rightarrow \rangle$ is a type structure over the filter λ-lattice. Moreover it satisfies the following equations:*
(i) $\chi_\omega = \mathcal{F}(Type)$; *(iii)* $\chi_{\sigma\wedge\tau} = \chi_\sigma \cap \chi_\tau$;
(ii) $\chi_{\sigma\to\tau} = \chi_\sigma \Rightarrow \chi_\tau$; *(iv)* $\chi_{\sigma\vee\tau} = \chi_\sigma \overline{\cup} \chi_\tau = \{f \cap f' \mid f \in \chi_\sigma, f' \in \chi_\tau\}$.

As for system $\mathcal{B}$, the immediate consequence of Theorem 55 and of Proposition 57 is completeness. Redefining $\models$ for λ-lattices in the same way as it has been defined for semimodels in 35, this is stated as follows.

Corollary 58 (Completeness of $\mathcal{L}$). $\Gamma \vdash_{\mathcal{L}} M{:}\sigma \ \Leftrightarrow\ \Gamma \models M{:}\sigma$.

The filter λ-lattice naturally induces a preorder on terms.

Definition 59. $M \sqsubseteq^{\mathcal{L}} N \Leftrightarrow_{def} \forall \rho.\ [\![M]\!]^{\mathcal{L}}_{\rho} \subseteq [\![N]\!]^{\mathcal{L}}_{\rho}$.

We state some (in)-equations which show that $\sqsubseteq^{\mathcal{L}}$ discriminates terms which are equated by $\sqsubseteq^{\mathcal{O}}$. This implies that the filter λ-lattice is not fully abstract with respect to the contextual semantics.

Proposition 60. *The following (in)-equations hold:*

(i) $(\lambda x.M)N \simeq^{\mathcal{L}} M[N/x]$;	*(vii)* $\lambda x.(M \| N) \simeq^{\mathcal{L}} \lambda x.M \| \lambda x.N$;
(ii) $(M+N)L \simeq^{\mathcal{L}} ML + NL$;	*(viii)* $M+N \sqsubseteq^{\mathcal{L}} M, N$;
(iii) $L(M+N) \sqsubseteq^{\mathcal{L}} LM + LN$;	*(ix)* $L \sqsubseteq^{\mathcal{L}} M, N \Rightarrow L \sqsubseteq^{\mathcal{L}} M+N$;
(iv) $(M \| N)L \simeq^{\mathcal{L}} ML \| NL$;	*(x)* $M, N \sqsubseteq^{\mathcal{L}} M \| N$;
(v) $LM \| LN \sqsubseteq^{\mathcal{L}} L(M \| N)$;	*(xi)* $M, N \sqsubseteq^{\mathcal{L}} L \Rightarrow M \| N \sqsubseteq^{\mathcal{L}} L$.
(vi) $\lambda x.(M+N) \sqsubseteq^{\mathcal{L}} \lambda x.M + \lambda x.N$;	

where the inequalities (iii), (v) and (vi) are in general proper.

Proof. Points $(i), (ii), (iv), (viii), (ix), (x)$ and (xi) hold by definition of λ-lattice. For the other points, the positive statements are easy consequences of Lemma 48.

The examples given in the proof of Proposition 15 show that the inequalities *(iii)* and *(v)* are proper. Indeed we have that both $\Delta M + \Delta N$ and $(T+R)(\mathbf{I}\|\mathbf{K})$ have type $\sigma \wedge (\sigma \to \tau) \to \tau$. On the contrary, ω is the only type which can be deduced for $\Delta(M+N)$ and for $(T+R)\mathbf{I}\|(T+R)\mathbf{K}$. To prove that the inequality *(vi)* is proper we have for example $\vdash_{\mathcal{L}} \lambda x.x + \lambda x.xx : (\mu \to \mu) \vee (\sigma \wedge (\sigma \to \tau) \to \tau)$, but this type cannot be deduced for $\lambda x.(x+xx)$. □

Notice that the filter model turns out to be a (properly) semilinear applicative structure as defined in [29, 30], because of 60*(ii)* and *(iii)*. This was not true for the set semimodel. It is worth to stress that, without the union type constructor, this cannot be achieved (see [1]). From this fact and from Proposition 38 it is also clear that the theories induced by $\simeq^{\mathcal{B}}$ and $\simeq^{\mathcal{L}}$ are incomparable.

6 Approximation Theorem and Full Abstraction

In this section we prove the main results of the present paper, i.e.:

- the filter λ-lattice is adequate with respect to the contextual semantics;
- the filter λ-lattice is fully abstract with respect to the capabilities semantics.

A main tool in these proofs is the notion of approximant. The first result essentially follows from the Approximation Theorem for the filter λ-lattice. For the second result we introduce a one-to-one correspondence between approximate normal forms (considered modulo $\sim$) and suitable pairs $<$ *basis, type* $>$ (where types are considered modulo $=$). This correspondence essentially shows that the discrimination power of approximants and types is the same.

6.1 The Approximation Theorem and The Adequacy for the Contextual Semantics

In this section we prove that the set of types which can be deduced for any term coincides with the union of the sets of types deducible for its approximants. Since in the filter λ-lattice these sets are thought of as the "meanings" of terms, this shows that the meaning of any term is the join of the meanings of its approximants.

Let us call $\mathcal{L}\Omega$ the type system resulting from $\mathcal{L}$ when subjects are from $\Lambda_{+\|\Omega}$. Since no explicit typing rule is added for the constant Ω, if $\Gamma \vdash_{\mathcal{L}\Omega} \Omega{:}\sigma$, then $\sigma = \omega$. Viceversa, a straightforward induction shows that, if A is an approximate normal form and $A \not\sim \Omega$, then there are a basis Γ and a type $\sigma \neq \omega$, such that $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$. All the properties of the system $\mathcal{L}$ proved in previous section extends easily to $\mathcal{L}\Omega$. So we will freely use them in the following proofs.

The Approximation Theorem is proved by means of a variant of Tait's "computability" technique. We define sets of "approximable" and "computable" terms (Definition 61). The computable terms are defined by induction on types, and every computable term is shown to be approximable (Lemma 64*(ii)*). Using induction on typings, we show that every term is computable for the appropriate type (Lemma 67).

Definition 61. We define two predicates $\mathsf{App}(\Gamma,\sigma,M)$ and $\mathsf{Comp}(\Gamma,\sigma,M)$ as follows:

(i) $\mathsf{App}(\Gamma,\sigma,M) \Leftrightarrow \exists A \in \mathcal{A}(M).\ \Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$;

(ii) *(a)* $\mathsf{Comp}(\Gamma,\omega,M)$ is always true;

(b) $\mathsf{Comp}(\Gamma,t,M) \Leftrightarrow \mathsf{App}(\Gamma,t,M)$;

(c) $\mathsf{Comp}(\Gamma,\sigma\to\tau,M) \Leftrightarrow \forall \Gamma',N.\mathsf{Comp}(\Gamma',\sigma,N) \Rightarrow \mathsf{Comp}(\Gamma \uplus \Gamma',\tau,MN)$;

(d) $\mathsf{Comp}(\Gamma,\sigma\wedge\tau,M) \Leftrightarrow \mathsf{Comp}(\Gamma,\sigma,M)$ and $\mathsf{Comp}(\Gamma,\tau,M)$;

(e) $\mathsf{Comp}(\Gamma,\sigma\vee\tau,M) \Leftrightarrow \mathsf{App}(\Gamma,\sigma\vee\tau,M)$.

We can easily prove that Comp agrees with some head reductions. More precisely we have:

Lemma 62. *Let M be a redex and N its immediate contractum. Then, for any Γ,σ,*

$$\mathsf{Comp}(\Gamma,\sigma,N\vec{L}) \Rightarrow \mathsf{Comp}(\Gamma,\sigma,M\vec{L})$$

where $\vec{L}$ is any vector of terms.

Proof. The proof is by induction on σ.
If $\sigma \equiv t$ or $\sigma \equiv \sigma_1 \vee \sigma_2$ the thesis follows immediately from 25*(iv)* since the hypothesis on M and N implies $M\vec{L} \xrightarrow{*}{}^{h} N\vec{L}$, so that $\mathcal{A}(N\vec{L}) \subseteq \mathcal{A}(M\vec{L})$.
If $\sigma \equiv \sigma_1 \wedge \sigma_2$ the thesis follows by induction.
If $\sigma \equiv \sigma_1 \to \sigma_2$, let P be such that $\mathsf{Comp}(\Gamma',\sigma_1,P)$ so that by definition $\mathsf{Comp}(\Gamma \uplus \Gamma',\sigma_2,N\vec{L}P)$. This implies by induction $\mathsf{Comp}(\Gamma \uplus \Gamma',\sigma_2,M\vec{L}P)$, so we can conclude $\mathsf{Comp}(\Gamma,\sigma,M\vec{L})$, by the arbitrariness of the term P. □

Really Comp is invariant under $=_a$, but we do not prove this, since we would need $M =_a N \Rightarrow \mathcal{A}(M) = \mathcal{A}(N)$ (see Remark 26).

We show some properties of types which are deducible for approximate normal forms.

Lemma 63. *Let $A, A' \in \mathcal{A}$.*

(i) $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$ *and* $A \preceq A' \Rightarrow \Gamma \vdash_{\mathcal{L}\Omega} A'{:}\sigma$.

(ii) *Let $z \notin FV(M)$ and suppose that z does not occur in the basis Γ.*
If $A \in \mathcal{A}(Mz)$, then

$$\Gamma, z{:}\sigma \vdash_{\mathcal{L}\Omega} A{:}\tau \Rightarrow \exists \widehat{A} \in \mathcal{A}(M).\ \Gamma \vdash_{\mathcal{L}\Omega} \widehat{A}{:}\sigma \to \tau.$$

Proof. *(i)* By induction on $\preceq$. The more interesting case is $A \equiv \lambda y.xA_1\ldots A_n y$ and $A' \equiv xA_1\ldots A_n$, where $y \notin FV(xA_1\ldots A_n)$. By 48*(ii)* $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$ implies $\Gamma \vdash_{\mathcal{L}\Omega} A{:} \bigwedge_{i=1}^m(\mu_i \to \nu_i)$, for some $\mu_1,\ldots,\mu_m,\nu_1,\ldots,\nu_m$ such that $\bigwedge_{i=1}^m(\mu_i \to \nu_i) \leq \sigma$. From $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\mu_i \to \nu_i$ by 48*(iii)* we have $\Gamma, y{:}\mu_i \vdash_{\mathcal{L}\Omega} xA_1\ldots A_n y{:}\nu_i$. Therefore by 48*(iv)* and 48*(i)* $\Gamma, y{:}\mu_i \vdash_{\mathcal{L}\Omega} xA_1\ldots A_n{:}\mu_i \to \nu_i$ hold for $1 \leq i \leq m$. Since $y \notin FV(xA_1\ldots A_n)$, we can prove using $(\wedge\mathrm{I})$ and $(\leq)$ that $\Gamma \vdash_{\mathcal{L}\Omega} A'{:}\sigma$.

(ii) $\mathcal{A}(M)$ is the downward closure of

$$\mathcal{A}'(M) = \{\phi(H) \mid H \in \mathcal{C}(M)\}$$

with respect to $\preceq$. It follows that, by *(i)*, it suffices to show the thesis when $A \in \mathcal{A}'(Mz)$.
If $A \in \mathcal{A}'(Mz)$ then, for some H, H',

$$A \equiv \phi(H') \text{ and } Mz \xrightarrow{*}{}^{h} H \xrightarrow{*}_{a} H'.$$

The proof is by induction on the length k of the reduction $\xrightarrow{*}{}^{h}$. If $k = 0$ then $Mz \equiv H \equiv xM_1 \ldots M_n z$. Hence $H' \equiv xM'_1 \ldots M'_n z$ where $M'_i \xrightarrow{*}_a M_i$ for $i \leq n$. Therefore $A \equiv x\phi(M'_1)\ldots\phi(M'_n)z$, so that we take $\widehat{A} \equiv x\phi(M'_1)\ldots\phi(M'_n) \in \mathcal{A}'(M)$. We have $\Gamma \vdash_{\mathcal{L}\Omega} \widehat{A}{:}\,\sigma \to \tau$ using 48*(i)* and *(iv)*.
If $k > 0$, then

$$Mz \xrightarrow{*}{}^{h} M'z \longrightarrow^{h} L \xrightarrow{*}{}^{h} H \xrightarrow{*}_{a} H'$$

where $M \xrightarrow{*}{}^{h} M'$ and M', L have one of the following shapes:

(a) $M' \equiv \lambda x.P$ and $L \equiv P[z/x]$;
(b) $M' \equiv P + Q$ and $L \equiv Pz + Qz$;
(c) $M' \equiv P||Q$ and $L \equiv Pz||Qz$.

Case (a). Then $A \in \mathcal{A}'(P[z/x])$, which implies $\lambda z.A \in \mathcal{A}'(\lambda z.P[z/x])$. Now $\lambda z.P[z/x] \equiv \lambda x.P$ since by hypothesis $z \notin FV(P)$. From $\Gamma, z{:}\,\sigma \vdash_{\mathcal{L}\Omega} A{:}\,\tau$ we derive by $(\to$ I) $\Gamma \vdash_{\mathcal{L}\Omega} \lambda z.A{:}\,\sigma \to \tau$. So we can choose $\widehat{A} \equiv \lambda z.A$, since $\mathcal{A}'(\lambda x.P) \subseteq \mathcal{A}'(M)$ by 25*(iv)*.

Case (b). In this case $H \equiv H_1 + H_2$, $H' \equiv H'_1 + H'_2$, and $Pz \xrightarrow{*}{}^{h} H_1 \xrightarrow{*}_a H'_1$, $Qz \xrightarrow{*}{}^{h} H_2 \xrightarrow{*}_a H'_2$. Moreover

$$A \equiv \phi(H'_1) + \phi(H'_2),$$

where $\phi(H'_1) \in \mathcal{A}'(Pz)$ and $\phi(H'_2) \in \mathcal{A}'(Qz)$. Now $\Gamma, z{:}\,\sigma \vdash_{\mathcal{L}\Omega} A{:}\,\tau$ implies, by Lemma 48*(v)*, $\Gamma, z{:}\,\sigma \vdash_{\mathcal{L}\Omega} \phi(H'_i){:}\,\tau$ for $i = 1, 2$. Notice that the length of the reductions $Pz \xrightarrow{*}{}^{h} H_1, Qz \xrightarrow{*}{}^{h} H_2$ is lower than k. Then by induction there are $A_1 \in \mathcal{A}'(P)$ and $A_2 \in \mathcal{A}'(Q)$ such that $\Gamma \vdash_{\mathcal{L}\Omega} A_i{:}\,\sigma \to \tau$, for $i = 1, 2$ hence

$$\Gamma \vdash_{\mathcal{L}\Omega} A_1 + A_2{:}\,\sigma \to \tau.$$

Therefore we can choose $\widehat{A} \equiv A_1 + A_2$; in fact $\widehat{A} \in \mathcal{A}'(M)$, since $M \xrightarrow{*}{}^{h} P + Q$.

Case (c). Similar to case (b), using Lemma 48*(vi)* and $M \xrightarrow{*}{}^{h} P||Q$. □

We can now show that computability implies approximability.

Lemma 64. *For all Γ, σ, $\vec{L}$ and M:*

(i) $\mathsf{App}(\Gamma, \sigma, x\vec{L}) \Rightarrow \mathsf{Comp}(\Gamma, \sigma, x\vec{L})$;
(ii) $\mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{App}(\Gamma, \sigma, M)$.

Proof. (i) and (ii) can be simultaneously proved by induction on σ. We show (ii) in the case $\sigma \equiv \sigma_1 \to \sigma_2$, only.

Let $\Gamma' = \Gamma, z{:}\,\sigma_1$ where $z \notin FV(M)$ and suppose $\mathsf{Comp}(\Gamma, \sigma_1 \to \sigma_2, M)$; then

$\mathsf{Comp}(\{z{:}\,\sigma_1\}, \sigma_1, z)$	by *(i)*
$\Rightarrow \mathsf{Comp}(\Gamma', \sigma_2, Mz)$	by definition
$\Rightarrow \mathsf{App}(\Gamma', \sigma_2, Mz)$	by induction
$\Rightarrow \exists A \in \mathcal{A}(M).\ \Gamma \vdash_{\mathcal{L}\Omega} A{:}\,\sigma_1 \to \sigma_2$	by Lemma 63*(ii)*.

□

The following two Lemmas state that computability agrees with the typing rules $(\leq)$, $(+\ \mathrm{I})$ and $(\|\ \mathrm{I})$.

Lemma 65. *For all σ and τ:*

(i) $\sigma \leq \tau \Rightarrow \forall \Gamma, M.\ \mathsf{App}(\Gamma, \sigma, M) \Rightarrow \mathsf{App}(\Gamma, \tau, M)$;

(ii) $\sigma \leq \tau \Rightarrow \forall \Gamma, M.\ \mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{Comp}(\Gamma, \tau, M)$.

Proof. If $A \in \mathcal{A}(M)$ is such that $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$ then by rule $(\leq)$ $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\tau$, hence (i).

(ii) is easily proved, using *(i)* and Lemma 64*(ii)*, by induction on any standard axiomatic presentation of $\leq$. In particular, for the basic case $\sigma \leq \sigma \vee \tau$ we have:

$$\begin{array}{rcll} \mathsf{Comp}(\Gamma, \sigma, M) & \Rightarrow & \mathsf{App}(\Gamma, \sigma, M) & \text{by Lemma 64}\textit{(ii)} \\ & \Rightarrow & \mathsf{App}(\Gamma, \sigma \vee \tau, M) & \text{by } \textit{(i)} \\ & \Rightarrow & \mathsf{Comp}(\Gamma, \sigma \vee \tau, M) & \text{by definition.} \end{array}$$

□

Lemma 66. *For all Γ, σ, τ and terms M, N:*

(i) $\mathsf{Comp}(\Gamma, \sigma, M)$ *and* $\mathsf{Comp}(\Gamma, \tau, N) \Rightarrow \mathsf{Comp}(\Gamma, \sigma \vee \tau, M + N)$;

(ii) $\mathsf{Comp}(\Gamma, \sigma, M) \Rightarrow \mathsf{Comp}(\Gamma, \sigma, M\|N)$.

Proof. *(i)* By Lemma 64*(ii)*, the hypothesis implies $\mathsf{App}(\Gamma, \sigma, M)$ and $\mathsf{App}(\Gamma, \tau, N)$, that is, for some $A \in \mathcal{A}(M)$ and $A' \in \mathcal{A}(N)$, $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$ and $\Gamma \vdash_{\mathcal{L}\Omega} A'{:}\tau$. This implies, by rules $(\leq)$ and $(+\ \mathrm{I})$, that $\Gamma \vdash_{\mathcal{L}\Omega} A + A'{:}\sigma \vee \tau$. Since $A + A' \in \mathcal{A}(M + N)$, it follows that $\mathsf{App}(\Gamma, \sigma \vee \tau, M + N)$, hence the thesis by definition.

(ii) By induction on σ. If σ has the shape t or $\sigma_1 \vee \sigma_2$, then $\mathsf{Comp}(\Gamma, \sigma, M)$ implies (by definition) $\mathsf{App}(\Gamma, \sigma, M)$, that is, for some $A \in \mathcal{A}(M)$, $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\sigma$. Hence, by rule $(\|\ \mathrm{I})$, $\Gamma \vdash_{\mathcal{L}\Omega} A\|\Omega{:}\sigma$. Since $A\|\Omega \in \mathcal{A}(M\|N)$ for any N, we conclude that $\mathsf{App}(\Gamma, \sigma, M\|N)$ holds. This implies the thesis.

If $\sigma \equiv \sigma_1 \wedge \sigma_2$, then the thesis is an immediate consequence of the induction hypothesis.

Finally, if $\sigma \equiv \sigma_1 \to \sigma_2$, let P be any term such that $\mathsf{Comp}(\Gamma', \sigma_1, P)$, so that by definition $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP)$. By induction,

$$\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP\|Q),$$

for any Q, hence for any N we can take $Q \equiv NP$ so that

$$\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, MP\|NP).$$

Lemma 62 implies $\mathsf{Comp}(\Gamma \uplus \Gamma', \sigma_2, (M\|N)P)$, so we can conclude:

$$\mathsf{Comp}(\Gamma, \sigma_1 \to \sigma_2, M\|N).$$

□

Lemma 67. *Let $\Gamma = \{x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n\}$ and $\Gamma \vdash_{\mathcal{L}} M{:}\tau$.*

Assume that, for each $i \leq n$, $\mathsf{Comp}(\Gamma_i, \sigma_i, N_i)$; then, taking $\Gamma' = \biguplus_{i=1}^{n} \Gamma_i$,

$$\mathsf{Comp}(\Gamma', \tau, M[N_1/x_1, \ldots, N_n/x_n]).$$

Proof. By induction on the derivation of $\Gamma \vdash_{\mathcal{L}} M{:}\tau$.

Cases (Ax) and (ω) are immediate.

Cases ($\rightarrow$ E) and ($\wedge$ I) follow by induction. Cases (+ I) and ($\leq$) follow from the induction hypothesis and Lemmas 66*(i)* and 65*(ii)* respectively.
If we are in case ($||$ I), then $M \equiv P||Q$ for some P and Q and, say, $\Gamma \vdash_{\mathcal{L}} P:\tau$ has been derived. From the induction hypothesis, $\mathsf{Comp}(\Gamma', \tau, P[\vec{N}/\vec{x}])$, so that by Lemma 66*(ii)*,

$$\mathsf{Comp}(\Gamma', \tau, P[\vec{N}/\vec{x}]||Q[\vec{N}/\vec{x}]),$$

i.e. $\mathsf{Comp}(\Gamma', \tau, (P||Q)[\vec{N}/\vec{x}])$.
Finally, in case ($\rightarrow$ I) suppose that $M \equiv \lambda y.P$, $\tau \equiv \tau_1 \rightarrow \tau_2$ and $\Gamma, y:\tau_1 \vdash P:\tau_2$ has been derived. Now, if $\mathsf{Comp}(\Gamma'', \tau_1, Q)$, from the induction hypothesis

$$\mathsf{Comp}(\Gamma' \uplus \Gamma'', \tau_2, P[Q/y, \vec{N}/\vec{x}]).$$

There is no theoretical loss in assuming that $y \notin FV(\vec{N})$ so that

$$P[Q/y, \vec{N}/\vec{x}] \equiv P[\vec{N}/\vec{x}][Q/y] \text{ and } (\lambda y.P[\vec{N}/\vec{x}])Q \equiv ((\lambda y.P)[\vec{N}/\vec{x}])Q.$$

By 62, it follows that $\mathsf{Comp}(\Gamma' \uplus \Gamma'', \tau_2, ((\lambda y.P)[\vec{N}/\vec{x}])Q)$, and hence

$$\mathsf{Comp}(\Gamma', \tau_1 \rightarrow \tau_2, (\lambda y.P)[\vec{N}/\vec{x}])$$

being the computable term Q arbitrary. □

Theorem 68 (Approximation Theorem). *For any term M, basis Γ and type σ:*

$$\Gamma \vdash_{\mathcal{L}} M:\sigma \Leftrightarrow \exists A \in \mathcal{A}(M).\ \Gamma \vdash_{\mathcal{L}\Omega} A:\sigma.$$

Proof. ($\Rightarrow$) Since, for any variable x and type τ, $\mathsf{App}(\{x:\tau\}, \tau, x)$ holds, then by Lemma 64*(i)*, $\mathsf{Comp}(\{x:\tau\}, \tau, x)$ holds. Taking in Lemma 67 the identical substitution, the hypothesis implies $\mathsf{Comp}(\Gamma, \sigma, M)$, and the thesis follows from Lemma 64*(ii)*.

($\Leftarrow$) Easy from subject conversion (Theorem 49) and the definition of $\mathcal{A}$. □

From the Approximation Theorem it follows that any term which is typeable with a type $\neq \omega$ has an approximant which differs from Ω, i.e. it is solvable. Viceversa, by Proposition 27*(ii)* any solvable term has an approximant different from Ω and therefore it can be typed with a type $\neq \omega$.

Corollary 69.

$$\mathsf{SOL} = \{M \in \Lambda_{+||} \mid \exists \Gamma, \sigma \neq \omega.\ \Gamma \vdash_{\mathcal{L}} M:\sigma\}.$$

The Approximation Theorem is useful to state properties of the precongruence induced on terms by the filter λ-lattice. In fact we immediately have that the filter λ-lattice is adequate with respect to the observational semantics based on contexts.

Theorem 70 (First Adequacy Theorem). *The filter λ-lattice is adequate for the contextual theory based on solvability, i.e.:*

$$M \sqsubseteq^{\mathcal{L}} N \Rightarrow M \sqsubseteq^{\mathcal{O}} N.$$

Proof. Since $\sqsubseteq^{\mathcal{L}}$ is a precongruence, we immediately have that

$$M \sqsubseteq^{\mathcal{L}} N \Rightarrow \forall C[\,].\ C[M] \sqsubseteq^{\mathcal{L}} C[N].$$

It follows that, by Corollary 69,

$$\begin{aligned} C[M] \in \mathsf{SOL} &\Rightarrow \exists \Gamma, \sigma \neq \omega.\ \Gamma \vdash_{\mathcal{L}} C[M]:\sigma \\ &\Rightarrow \exists \Gamma, \sigma \neq \omega.\ \Gamma \vdash_{\mathcal{L}} C[N]:\sigma \\ &\Rightarrow C[N] \in \mathsf{SOL}. \end{aligned}$$

□

6.2 Principal Pairs and Full Abstraction for the Capability Semantics

To prove adequacy for the semantics based on capabilities and approximants, a suitable extension of the notion of principal type scheme (as given in [17, 40, 9]) is in order. Since we need to consider open terms, we introduce the notion of *principal pair* consisting of a type and a basis. Such a notion is based on a stratification of the set of approximate normal forms, to be compared with the stratification of *Type* introduced in Definition 40.

Definition 71 (Stratification of $\mathcal{A}$). Let us define three subsets $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ of $\mathcal{A}$ recursively:

- $\Omega \in \mathcal{A}_2$;
- $A \in \mathcal{A}_1 \Rightarrow \lambda x.A \in \mathcal{A}_0$;
- $m \geq 0, A_1, \ldots, A_m \in \mathcal{A}_2 \Rightarrow xA_1 \ldots A_m \in \mathcal{A}_0$ (the λ-free approximate normal forms);
- $n \geq 1, A_1, \ldots, A_n \in \mathcal{A}_0 \Rightarrow A_1 + \ldots + A_n \in \mathcal{A}_1$;
- $n \geq 1, A_1, \ldots, A_n \in \mathcal{A}_1 \Rightarrow A_1 || \ldots || A_n \in \mathcal{A}_2$.

Taking $n = 1$ in the clauses above, one sees that $\mathcal{A}_0 \subseteq \mathcal{A}_1 \subseteq \mathcal{A}_2$, and such inclusions are clearly proper. Over each of these sets we introduce a preorder.

Definition 72. $\preceq_i \subseteq \mathcal{A}_i \times \mathcal{A}_i$ is the least preorder such that:

($\preceq_0$) $A \preceq_0 A'$ if and only if one of the following holds:

- $A \equiv \lambda x.B, A' \equiv \lambda x.B'$ and $B \preceq_1 B'$;
- $A \equiv xB_1 \ldots B_n, A' \equiv xB'_1 \ldots B'_n$ and $\forall i \leq n. B_i \preceq_2 B'_i$;
- A' is λ-free, $x \notin FV(A')$ and $A \preceq_0 \lambda x.A'x$.

($\preceq_1$) $A_1 + \ldots + A_n \preceq_1 B_1 + \ldots + B_m \;\Leftrightarrow\; \forall j \leq m\; \exists i \leq n.\; A_i \preceq_0 B_j$.

($\preceq_2$) $A \preceq_2 A'$ if and only if one of the following holds:

- $A \equiv \Omega$;
- $A \equiv B_1 || \ldots || B_n, A' \equiv B'_1 || \ldots || B'_m$ and $\forall i \leq n\; \exists j \leq m.\; B_i \preceq_1 B'_j$.

As in the case of types, for each approximate normal form we can find an equivalent element of $\mathcal{A}_2$. The following definition has to be compared with 43.

Notation. In writing $A^* \equiv ||_{i\in I} A_i$ we assume that $A_i \in \mathcal{A}_1$ for all $i \in I$.

Definition 73. Let $(\;)^* : \mathcal{A} \rightarrow \mathcal{A}_2$ be defined by:

- $\Omega^* = \Omega$
- $(xA_1 \ldots A_n)^* = xA_1^* \ldots A_n^* \qquad (n \geq 0)$
- $(\lambda x.A)^* = \begin{cases} \lambda x.A_1 || \ldots || \lambda x.A_n & \text{if } A^* = A_1 || \ldots || A_n \text{ and } A^* \not\equiv \Omega \qquad (n \geq 1) \\ \Omega & \text{otherwise} \end{cases}$
- $(A + A')^* = \begin{cases} ||_{i\in I,\, j\in J}(B_i + B'_j) & \text{if } A^* = ||_{i\in I} B_i,\ A^* \not\equiv \Omega \text{ and } A'^* = ||_{j\in J} B'_j,\ A'^* \not\equiv \Omega \\ \Omega & \text{otherwise} \end{cases}$
- $(A||A')^* = \begin{cases} A^* & \text{if } A'^* \equiv \Omega \\ A'^* & \text{if } A^* \equiv \Omega \\ A^* || A'^* & \text{otherwise.} \end{cases}$

The proof of the following proposition is analogous to the proof of Proposition 44.

Proposition 74. *For all $A, A' \in \mathcal{A}$:*

(i) $A \sim A^*$;

(ii) $A, A' \in \mathcal{A}_i, A \preceq_i A' \Rightarrow A \preceq A'$ *for* $i = 0, 1, 2$;

(iii) $A \preceq A' \Rightarrow A^* \preceq_2 A'^*$.

The following definition of principal pair is a generalization to our calculus of that one given in [17], [40], and [9], where it was used to prove the principal type property for various intersection type disciplines.

Let *Basis* be the set of all bases and $TV(<\Gamma;\sigma>)$ be the set of type variables which occur in Γ or in σ.

Definition 75.

(i) The mapping $pp : \mathcal{A}_2 \rightarrow Basis \times T_2$ is inductively defined by:

(a) $pp(\Omega) = <\emptyset;\omega>$;

(b) if $pp(A_i) = <\Gamma_i;\sigma_i>$, $TV(<\Gamma_i;\sigma_i>) \cap TV(<\Gamma_j;\sigma_j>) = \emptyset$ for $1 \leq i \neq j \leq n$ and t is fresh, then

$$pp(xA_1 \ldots A_n) = <(\biguplus_{i\leq n} \Gamma_i) \uplus \{x{:}\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow t\}; t> \ (n \geq 0);$$

(c) if $pp(A) = <\Gamma, x : \tau; \sigma>$, then

$$pp(\lambda x.A) = <\Gamma; \tau \rightarrow \sigma>;$$

(d) if $pp(A) = <\Gamma;\sigma>$ and $x \notin FV(\Gamma)$, then

$$pp(\lambda x.A) = <\Gamma; \omega \rightarrow \sigma>;$$

(e) if $pp(A_i) = <\Gamma_i;\sigma_i>$ $(i = 1, 2)$ and $TV(<\Gamma_1;\sigma_1>) \cap TV(<\Gamma_2;\sigma_2>) = \emptyset$, then

$$pp(A_1 + A_2) = <\Gamma_1 \uplus \Gamma_2; \sigma_1 \vee \sigma_2>;$$

(f) if $pp(A_i) = <\Gamma_i;\sigma_i>$ $(i = 1, 2)$ and $TV(<\Gamma_1;\sigma_1>) \cap TV(<\Gamma_2;\sigma_2>) = \emptyset$, then

$$pp(A_1 || A_2) = <\Gamma_1 \uplus \Gamma_2; \sigma_1 \wedge \sigma_2>.$$

(ii) The set Π of principal pairs is the range of the mapping pp.

(iii) A type σ is *principal* iff $<\Gamma,\sigma> \in \Pi$ for some basis Γ. A basis Γ is *principal* iff $<\Gamma,\sigma> \in \Pi$ for some type σ.

To build a unique principal pair, in clause 75*(i) (b)* we assume to pick up fresh type variables in a deterministic way.

For example we have

$$pp(xyy + (\lambda z.y||\lambda z.z)) = <x : t_1 \rightarrow t_2 \rightarrow t_3, y : t_1 \wedge t_2 \wedge t_4 \ ; \ t_3 \vee ((\omega \rightarrow t_4) \wedge (t_5 \rightarrow t_5))>.$$

From the definition it follows immediately that the principal pair of an approximate normal form can be deduced for it. Moreover it is easy to prove that the mapping pp agrees with the stratification of types and approximate normal forms.

Proposition 76. *If* $pp(A) = <\Gamma;\sigma>$ *then* $\Gamma \vdash_{\mathcal{L}\Omega} A : \sigma$ *and* $A \in \mathcal{A}_i$ *iff* $\sigma \in T_i$ *where* $i = 0, 1, 2$.

Π turns out to be a very restricted set with closure properties which follow easily from its definition.

Proposition 77. *Let* $<\Gamma;\sigma>\in\Pi$.

(i) *Each type variable occurs exactly twice in* Γ,σ.

(ii) *All types which occur in a principal basis belong to* T_2. *Moreover they are intersections of arrow types belonging to* T_0 *and terminating with a type variable.*

(iii) *If* $x{:}\tau_1\rightarrow\ldots\rightarrow\tau_n\rightarrow\mu\in\Gamma$, *then for all* $1\leq i\leq n$ *there is* $\Gamma_i\Subset\Gamma$ *such that* $<\Gamma_i;\tau_i>\in\Pi$.

(iv) *If* $\sigma\equiv\mu\rightarrow\tau$, *then* $<\Gamma,x:\mu\,;\tau>\in\Pi$ *for all variables* x.

(v) *If* $\sigma\equiv\sigma_1\vee\sigma_2$ *or* $\sigma\equiv\sigma_1\wedge\sigma_2$, *then there are* $\Gamma_1,\Gamma_2\Subset\Gamma$ *such that* $<\Gamma_i;\sigma_i>\in\Pi\ \ (i=1,2)$.

The types which can be deduced for a variable from a principal basis are of limited shape.

Lemma 78. *Let* Γ *be a principal basis.*

(i) *If* $\tau\in T_1$ *and* $\Gamma\vdash x{:}\tau$, *then* $\Gamma(x)=\mu\wedge\nu$ *for some* μ,ν *such that* $\mu\in T_0$ *and* $\mu\leq_1\tau$.

(ii) *If* $\mu\vee\nu\in T_1$ *and* $\Gamma\vdash x{:}\tau_1\rightarrow\ldots\rightarrow\tau_n\rightarrow\mu\vee\nu$, *then either* $\Gamma\vdash x{:}\tau_1\rightarrow\ldots\rightarrow\tau_n\rightarrow\mu$ *or* $\Gamma\vdash x{:}\tau_1\rightarrow\ldots\rightarrow\tau_n\rightarrow\nu$ $(n\geq 0)$.

Proof. *(i)* Notice that $\tau\in T_1$ implies $\tau^*\equiv\tau$.

$\Gamma\vdash_{\mathcal{L}\Omega} x{:}\tau$
$\Rightarrow\Gamma(x)\leq\tau$ by Lemma 48*(i)*
$\Rightarrow\Gamma(x)\leq_2\tau$ by 77*(ii)* and Proposition 44*(iii)*
$\Rightarrow\exists\mu\in T_1,\nu.\ \Gamma(x)=\mu\wedge\nu$ and $\mu\leq_1\tau$ by Definition 42 since $\tau\in T_1$
$\Rightarrow\exists\mu\in T_0,\nu.\ \Gamma(x)=\mu\wedge\nu$ and $\mu\leq_1\tau$ by 77*(ii)*.

(ii) From *(i)* there are $\sigma_1\in T_0,\sigma_2$ such that $\Gamma(x)=\sigma_1\wedge\sigma_2$ and $\sigma_1\leq_1\tau_1^*\rightarrow\ldots\rightarrow\tau_n^*\rightarrow\mu\vee\nu$. Let $\sigma_1\equiv\xi_1\rightarrow\ldots\rightarrow\xi_n\rightarrow\xi$, where $\xi\in T_0$ by 77*(ii)*. Then $\xi\leq_1\mu\vee\nu$ which implies, by Definition 42, either $\xi\leq_1\mu$ or $\xi\leq_1\nu$. □

The principal pair carries out the same information of the corresponding approximate normal form. This implies that $pp(A)$ can be deduced only for approximate normal forms which are better than A according to the preorder $\preceq$. The proof of this fact will be done in Lemma 81 using some preliminary properties (Lemmas 79 and 80).

Lemma 79. *Let* $A\in\mathcal{A}$, Γ *be a principal basis, and* $\Gamma\vdash_{\mathcal{L}\Omega}A{:}\sigma$.

(i) $\sigma\in T_1$ *implies* $\exists A'\in\mathcal{A}_1,A''\in\mathcal{A}.\ A\sim A'||A''$ *and* $\Gamma\vdash_{\mathcal{L}\Omega}A'{:}\sigma$.

(ii) $A\in\mathcal{A}_0$, *and* $\sigma\equiv\sigma_1\vee\sigma_2\in T_1$ *imply either* $\Gamma\vdash_{\mathcal{L}\Omega}A{:}\sigma_1$ *or* $\Gamma\vdash_{\mathcal{L}\Omega}A{:}\sigma_2$.

Proof. *(i)* If $A\in\mathcal{A}_1$ it is trivial choosing $A'\equiv A$ and $A''\equiv\Omega$. Otherwise, let $A\equiv A_1||\ldots||A_m$, where $A_i\in\mathcal{A}_1$ $(1\leq i\leq m)$. Then $\Gamma\vdash_{\mathcal{L}\Omega}A{:}\sigma\Rightarrow\exists\tau_1,\ldots,\tau_m.\ \Gamma\vdash_{\mathcal{L}\Omega}A_i{:}\tau_i$ and $\tau_1\wedge\ldots\wedge\tau_m\leq\sigma$ by Lemma 48*(v)*.

Let $\tau_i^*=\bigwedge_{l\in L}\nu_{i,l}$ (where L depends on i). Then $\tau_1\wedge\ldots\wedge\tau_m\leq\sigma$ and $\sigma\in T_1$ imply that there exist i,l such that $\nu_{i,l}\leq_1\sigma$, by Definition 42 and Proposition 44. Hence $\Gamma\vdash_{\mathcal{L}\Omega}A_i{:}\sigma$.

(ii) By cases on $A\in\mathcal{A}_0$.

- $A\equiv\Omega$ is trivial.

- $A \equiv xA_1 \ldots A_m$ $(m \geq 0)$ implies by 48*(iv)* $\Gamma \vdash_{\mathcal{L}\Omega} x{:}\tau_1 \rightarrow \ldots \rightarrow \tau_m \rightarrow \sigma$ for some $\tau_1, \ldots, \tau_m$, so the result follows by 78*(ii)*.
- $A \equiv \lambda x.A'$ implies, by Lemma 48*(ii)*, $\tau_1 \wedge \ldots \wedge \tau_m \leq \sigma$ and $\Gamma \vdash_{\mathcal{L}\Omega} A{:}\tau_j$ $(j \leq m)$ for some arrow types $\tau_1, \ldots, \tau_m$. Let $\tau_j^* = \bigwedge_{l\in L}(\mu_{j,l} \rightarrow \nu_{j,l})$ (where L depends on j). We have by Definition 42 and Proposition 44 that $\mu_{j,l} \rightarrow \nu_{j,l} \leq_1 \sigma$ for some j, l , since $\sigma \in T_1$. Therefore if $\sigma \equiv \sigma_1 \vee \sigma_2$ we have by Definition 42 and by Proposition 44 $\mu_{j,l} \rightarrow \nu_{j,l} \leq_1 \sigma_1$ or $\mu_{j,l} \rightarrow \nu_{j,l} \leq_1 \sigma_2$. □

Lemma 80. *Let $A \in \mathcal{A}$, Γ, Γ' be principal basis and τ be a principal type such that $\Gamma' \Subset \Gamma$ and $< \Gamma'; \tau > \in \Pi$. Then $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau$ implies $\Gamma' \vdash_{\mathcal{L}\Omega} A : \tau$.*

Proof. We prove a more general statement, i.e.:

Let $\Gamma, \Gamma', \Gamma''$ be principal basis and τ be a principal type such that $\Gamma'' \Subset \Gamma' \Subset \Gamma$ and $< \Gamma''; \tau > \in \Pi$. Then $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau$ implies $\Gamma' \vdash_{\mathcal{L}\Omega} A : \tau$.

The proof is by a principal induction on A and a secondary induction on τ.
The case $A \equiv \Omega$ is immediate.
The case $\tau \in T_2 - T_1$ follows easily by the secondary induction. In fact if $\tau \equiv \tau_1 \wedge \tau_2$, then $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau$ implies both $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau_1$ and $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau_2$. Moreover by 77*(v)* there are $\Gamma_1, \Gamma_2 \Subset \Gamma'$ such that $< \Gamma_i, \tau_i > \in \Pi$ $(i = 1, 2)$.

$A \equiv xA_1 \ldots A_n$ and $n \geq 0$ implies by 48*(iv)* $\Gamma \vdash_{\mathcal{L}\Omega} x{:}\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow \tau$ for some $\sigma_1, \ldots, \sigma_n$, such that $\Gamma \vdash_{\mathcal{L}\Omega} A_i{:}\sigma_i$ for all $i \leq n$. By 78*(i)* $\Gamma(x) = \mu \wedge \nu$ for some μ, ν such that $\mu \in T_0$ and $\mu \leq_1 \sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow \tau$. Let $\tau \equiv \sigma_{n+1} \rightarrow \ldots \rightarrow \sigma_{n+m} \rightarrow \tau'$ $(m \geq 0)$, where either τ' is a type variable, say $\tau' \equiv t$, or $\tau' \in T_1 - T_0$. Then $\mu \equiv \sigma'_1 \rightarrow \ldots \rightarrow \sigma'_{n+m} \rightarrow t$ with $t \leq_1 \tau'$ and $\sigma_i \leq_2 \sigma'_i$ for $i \leq n+m$ by Definition 42. If $\tau' \not\equiv t$ by 42 we have that $\tau' = t \vee \tau''$ for some τ''. In both cases the hypothesis $< \Gamma''; \tau > \in \Pi$ assures us that t must occur in $\Gamma'' \Subset \Gamma'$. Therefore $\Gamma'(x) = \mu \wedge \nu'$ for some ν' and we have $\Gamma' \vdash_{\mathcal{L}\Omega} x : \sigma'_1 \rightarrow \ldots \rightarrow \sigma'_{n+m} \rightarrow \tau'$. $\Gamma \vdash_{\mathcal{L}\Omega} A_i{:}\sigma_i$ implies $\Gamma \vdash_{\mathcal{L}\Omega} A_i{:}\sigma'_i$ by rule $(\leq)$. By 77*(iii)* there are $\Gamma_i \Subset \Gamma'$ such that $< \Gamma_i, \sigma'_i > \in \Pi$. Therefore by the principal induction $\Gamma' \vdash_{\mathcal{L}\Omega} A_i{:}\sigma'_i$ for all $i \leq n$. So we can conclude $\Gamma' \vdash_{\mathcal{L}\Omega} A : \tau$.

$A \equiv \lambda x.A'$.

$\tau \in T_0$. Let $\tau \equiv \tau_1 \rightarrow \tau_2$. By 77*(iv)* $< \Gamma'', x : \tau_1; \tau_2 > \in \Pi$. By Lemma 48*(iii)* $\Gamma, x : \tau_1 \vdash_{\mathcal{L}\Omega} A' : \tau_2$. Therefore by the principal or the secondary induction $\Gamma', x : \tau_1 \vdash_{\mathcal{L}\Omega} A' : \tau_2$. By rule $(\rightarrow \mathrm{I})$ we conclude $\Gamma' \vdash_{\mathcal{L}\Omega} A : \tau$.

$\tau \in T_1 - T_0$. Let $\tau \equiv \tau_1 \vee \tau_2$. By Lemma 79*(ii)* $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau_1$ or $\Gamma \vdash_{\mathcal{L}\Omega} A : \tau_2$. By 77*(v)* there are $\Gamma_1, \Gamma_2 \Subset \Gamma'$ such that $< \Gamma_i, \tau_i > \in \Pi$ $(i = 1, 2)$. Therefore the secondary induction applies.

$A \equiv A_1 + A_2$ implies by 48*(v)* $\Gamma \vdash_{\mathcal{L}\Omega} A_1{:}\tau$ and $\Gamma \vdash_{\mathcal{L}\Omega} A_2{:}\tau$. By the principal induction we have $\Gamma' \vdash_{\mathcal{L}\Omega} A_1{:}\tau$ and $\Gamma' \vdash_{\mathcal{L}\Omega} A_2{:}\tau$, so we can conclude $\Gamma' \vdash_{\mathcal{L}\Omega} A{:}\tau$ by rule $(+\,\mathrm{I})$.

$A \equiv A_1 || A_2$ implies by 48*(vi)* $\Gamma \vdash_{\mathcal{L}\Omega} A_1{:}\sigma_1$ and $\Gamma \vdash_{\mathcal{L}\Omega} A_2{:}\sigma_2$ for some σ_1, σ_2 such that $\sigma_1 \wedge \sigma_2 \leq \tau$. We need to consider only the case $\tau \in T_1$, therefore by 42 and 44 either $\sigma_1 \leq \tau$ or $\sigma_2 \leq \tau$. In the first case $\Gamma \vdash_{\mathcal{L}\Omega} A_1{:}\tau$, so by the principal induction $\Gamma' \vdash_{\mathcal{L}\Omega} A_1{:}\tau$. The second case is symmetric. □

Lemma 81 (Principal Pair Lemma).
If $A, A' \in \mathcal{A}$, $pp(A) =< \Gamma; \sigma >$ *and* $\Gamma \vdash_{\mathcal{L}\Omega} A' : \sigma$, *then* $A \preceq A'$.

Proof. By cases and then by induction on the structure of A. By hypothesis $A \in \mathcal{A}_2$.

Case $A \in \mathcal{A}_1$. In this case $\sigma \in T_1$, then by Lemma 79*(i)* there exists $B \in \mathcal{A}_1$ and some B' such that $A' \sim B || B'$ and $\Gamma \vdash_{\mathcal{L}\Omega} B{:}\sigma$. Let $B = B_1 + \ldots + B_n$ where $B_i \in \mathcal{A}_0$ $(1 \leq i \leq n)$; then $\Gamma \vdash_{\mathcal{L}\Omega} B_i{:}\sigma$ $(i \leq n)$ by Lemma 48*(v)*. We distinguish three subcases after the shape of σ.

Subcase $\sigma \equiv t$. In this case we have $A \equiv xA_1 \ldots A_m$ for some $A_1, \ldots, A_m$ $(m \geq 0)$. Moreover by 77*(i)* there is only one type in Γ which contains the type variable t; let $x{:}\tau_1 \to \ldots \to \tau_m \to t \in \Gamma$. Therefore we have by Definition 75*(i)(b)*:

$$\Gamma = (\biguplus_{j \leq m} \Gamma_j) \uplus \{x{:}\tau_1 \to \ldots \to \tau_m \to t\} \text{ and } \Gamma_j \vdash_{\mathcal{L}\Omega} A_j{:}\tau_j \ (j \leq m).$$

$\Gamma \vdash_{\mathcal{L}\Omega} B_i{:}t$ $(i \leq n)$ and $B_i \in \mathcal{A}_0$ imply by Lemma 48*(ii)* $B_i \equiv xC_{i,1} \ldots C_{i,m}$. Moreover using Lemma 48*(i)* and *(iv)* $\Gamma \vdash_{\mathcal{L}\Omega} C_{i,j}{:}\tau_j$ $(i \leq n, j \leq m)$. This implies by Lemma 80 $\Gamma_j \vdash_{\mathcal{L}\Omega} C_{i,j}{:}\tau_j$ $(i \leq n, j \leq m)$. So we have by induction $A_j \preceq C_{i,j}$ $(j \leq m)$. Therefore

$$A_j \preceq C_{i,j} \ (i \leq n, j \leq m) \Rightarrow A \preceq B_i \ (i \leq n) \Rightarrow A \preceq B \Rightarrow A \preceq A'.$$

Subcase $\sigma \equiv \tau \to \mu$. In this case $A \equiv \lambda x.A''$. If $B_i \equiv \lambda x.B'_i$, it is easy by induction. If B_i is a λ-free term, then also $\lambda z.B_i z \in \mathcal{A}_0$, where z is fresh, and $\Gamma \vdash_{\mathcal{L}\Omega} \lambda z.B_i z{:}\tau \to \mu$. We are in the previous case and we can prove $A \preceq_0 \lambda z.B_i z$, so we can conclude $A \preceq_0 B_i$.

Subcase $\sigma \equiv \tau_1 \vee \tau_2$. In this case we have $A \equiv A_1 + A_2$, $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $\Gamma_j \vdash_{\mathcal{L}\Omega} A_j{:}\tau_j$ $(j = 1, 2)$ by 75*(i)(d)*. $\Gamma \vdash_{\mathcal{L}\Omega} B_i{:}\sigma$ implies, by Lemma 79*(ii)*, $\exists l_i \leq 2$. $\Gamma \vdash_{\mathcal{L}\Omega} B_i{:}\tau_{l_i}$, since $B_i \in \mathcal{A}_0$. This implies by Lemma 80 $\Gamma_{l_i} \vdash_{\mathcal{L}\Omega} B_i{:}\tau_{l_i}$. By induction, $A_{l_i} \preceq B_i$, for all $i \leq n$, which implies $A \preceq B$, so we can conclude $A \preceq A'$.

Case $A \notin \mathcal{A}_1$. In this case $\sigma \equiv \tau_1 \wedge \tau_2$, $A \equiv A_1 || A_2$ $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $\Gamma_j \vdash_{\mathcal{L}\Omega} A_j{:}\tau_j$ $(j = 1, 2)$. By rule $(\leq)$ we have $\Gamma \vdash_{\mathcal{L}\Omega} A'{:}\tau_j$ $(j = 1, 2)$ and this implies by Lemma 80 $\Gamma_j \vdash_{\mathcal{L}\Omega} A'{:}\tau_j$ $(j = 1, 2)$. By induction $A_1 \preceq A'$ and $A_2 \preceq A'$, so we can conclude $A \preceq A'$. □

We are finally in place to prove:

Theorem 82 (Second Adequacy Theorem).
The filter λ-lattice is adequate for the semantics based on capabilities, i.e.:

$$M \sqsubseteq^{\mathcal{L}} N \Rightarrow M \sqsubseteq^{\mathcal{A}} N.$$

Proof. We prove $M \not\sqsubseteq^{\mathcal{A}} N \Rightarrow M \not\sqsubseteq^{\mathcal{L}} N$. By Proposition 25*(iii)*,

$$M \not\sqsubseteq^{\mathcal{A}} N \Rightarrow \exists A \in \mathcal{A}(M). A \notin \mathcal{A}(N).$$

Let $pp(A^*) =< \Gamma; \sigma >$; by the Approximation Theorem, $\Gamma \vdash_{\mathcal{L}} M{:}\sigma$. Assume now $\Gamma \vdash_{\mathcal{L}} N{:}\sigma$. Then, by the Approximation Theorem again, there exists $A' \in \mathcal{A}(N)$ such that $\Gamma \vdash_{\mathcal{L}\Omega} A'{:}\sigma$. Hence, by the Principal Pair Lemma, $A \preceq A'$ so that $A \in \mathcal{A}(N)$, which is absurd. It follows that $\Gamma \not\vdash_{\mathcal{L}} N{:}\sigma$, so we can conclude $M \not\sqsubseteq^{\mathcal{L}} N$. □

We immediately have

Theorem 83 (Full Abstraction Theorem). *The filter λ-lattice is fully abstract for the semantics based on capabilities, i.e.:*

$$M \sqsubseteq^{\mathcal{L}} N \Leftrightarrow M \sqsubseteq^{\mathcal{A}} N.$$

Proof. Immediate consequence of the Approximation Theorem 68 and of the Second Adequacy Theorem 82. □

From the Full Abstraction Theorem and the invariance of types under $=_a$ (Theorem 49), we have that the set of approximate normal forms is invariant under $=_a$ (see Remark 26).

By the Full Abstraction Theorem, in Proposition 60 we can replace $\sqsubseteq^{\mathcal{L}}$ by $\sqsubseteq^{\mathcal{A}}$.

Theorems 70 and 83 relate also the two operational semantics we considered: as expected $\sqsubseteq^{\mathcal{A}}$ turns out to be a refinement of $\sqsubseteq^{\mathcal{O}}$.

Acknowledgments

We are very grateful to Martin Abadi, Fabio Alessi, Steffen van Bakel, Furio Honsell and Jim Lipton for helpful discussions about the subject of this paper. We thank also the referees for their insighful comments.

References

1. M. Abadi, "A Semantics for Static Type Inference in a Non-Deterministic Language", *Info. and Comp.* 109, 1994, 300-306.
2. S. Abramsky, "On Semantic Foundations for Applicative Multiprogramming", ICALP'83, *LNCS* 154, Springer-Verlag, Berlin, 1983, 1-14.
3. S. Abramsky, "Domain Theory in Logical Form", *Ann. of Pure and Appl. Logics* 51, 1991, 1-77.
4. S. Abramsky, C.-H.L. Ong, "Full Abstraction in the Lazy Lambda Calculus", *Inf. and Comp.*105, 1993, 159-267.
5. F. Alessi, " Type Preorders", CAAP'94, *LNCS* 787, Springer-Verlag, Berlin, 1994, 37-51.
6. F. Alessi, M. Dezani-Ciancaglini, U. de'Liguoro, "Must and May convergency in Concurrent Lambda Calculus", MFCS'94, *LNCS* 841, Springer-Verlag, Berlin, 1994, 211-221.
7. E.A. Ashcroft, M.C.B. Hennessy, "A Mathematical Semantics for a Non-Deterministic Typed Lambda Calculus", *Theor. Comp. Sci.* 11, 1980, 227-245.
8. S.van Bakel, "Complete Restrictions of the Intersection Type Discipline", *Theor. Comp. Sci.* 102, 1992, 135-163.
9. S.van Bakel, "Principal Type Schemes for the Strict Type Assignment System", *J. Logic and Comp.* 3(6), 1993, 643-670.
10. S.van Bakel, "Essential Intersection Type Assignment System", FSTTCS'93, *LNCS* 761, Springer-Verlag, Berlin, 1993, 13-23.
11. F. Barbanera, M. Dezani-Ciancaglini, U. de'Liguoro, "Intersection and Union Types: Syntax and Semantics", *Info. and Comp.* 119(2), 1995, 202-230.
12. H.P. Barendregt, *The Lambda-Calculus: Its Syntax and Semantics*, North-Holland, Amsterdam, 1984.
13. H. Barendregt, M. Coppo, M. Dezani-Ciancaglini, "A Filter Lambda Model and the Completeness of Type Assignment", *J.Symbolic Logic* 48, 1983, 931-940.
14. G. Boudol, "Semantique Operationelle et Algebrique des Programmes Recursifs Non-Deterministes", Thèse d'Etat, Université de Paris VII, 1980.
15. G. Boudol, "Toward a Lambda-calculus for Concurrent and Communicating Systems", TAPSOFT'89, *LNCS* 351, Springer-Verlag, Berlin, 1989, 149-161.

16. G. Boudol, "A Lambda Calculus for (Strict) Parallel Functions", *Info. and Comp.* 108, 1994, 51-127.
17. M.Coppo, M. Dezani-Ciancaglini, B.Venneri, "Principal Type Schemes and λ-calculus Semantics", *To H.B.Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, New York, 1980, 535-560.
18. M.Coppo, A.Ferrari, "Type Inference, Abstract Interpretation and Strictness Analysis", *Theor. Comp. Sci.* 121, 1993, 113-144.
19. M. Dezani-Ciancaglini, U. de'Liguoro, A. Piperno, "Filter Models for a Parallel and Non-Deterministic λ-calculus", MFCS'93, *LNCS* 711, Springer-Verlag, Berlin, 1993, 403-412.
20. M. Dezani-Ciancaglini, U. de'Liguoro, A. Piperno, "Fully Abstract Semantics for Concurrent λ-calculus", TACS'94, *LNCS* 789, Springer-Verlag, Berlin, 1994, 16-35.
21. G.Gonthier, J.J. Lévy, P.A.Melliés, "An Abstract Standardization Theorem", *LICS '92*, IEEE Computer Soc. Press, Silver Spring, 1992, 72-81.
22. M. Hennessy, "A Fully Abstract Denotational Model for Higher-Order Processes", *Info. and Comp.* 112(1), 1994, 55-95.
23. R. Hindley, "The Completeness Theorem for Typing λ-terms", *Theor. Comp. Sci.* 22, 1983, 1-17.
24. R. Hindley,"The Simple Semantics for Coppo-Dezani-Sallé Type Assignment", Int. Symp. on Prog., *LNCS* 137, Springer-Verlag, Berlin, 1982, 212-226.
25. R. Hindley, G. Longo, "Lambda Calculus Models and Extensionality", *Z. Math. Logik* 26, 1980, 289-310.
26. T.P.Jensen, "Disjunctive Strictness Analisys", *LICS '92*, IEEE Comp. Soc. Press, Silver Spring, 1992, 174-185.
27. J.L. Krivine, *Lambda-calcul, types et modèles*, Masson, Paris 1990.
28. J.J. Lévy, "An Algebraic Interpretation of the $\lambda\beta K$-calculus and a Labelled λ-calculus", λ-calculus and Computer Science Theory, *LNCS* 37, 1975,147-165 .
29. U. de' Liguoro, "Non-deterministic Untyped λ-calculus", PhD Thesis, Un. di Roma I, 1991.
30. U. de' Liguoro, A. Piperno, "Must Preorder in Non-deterministic Untyped λ-calculus", *Info. and Comp.* 122(2), 1995, 149-177.
31. R. Milner, "Functions as Processes", *J. Math. Struc. in Comp. Sci.* 2(2), 1992, 119-142.
32. R. Milner, J.G. Parrow, D.J. Walker, "A Calculus of Mobile Processes, Part I" *Info. and Comp.* 100(1), 1992, 1-40.
33. J.H. Morris, *Lambda Calculus Models of Programming Languages*, Dissertation, M.I.T. 1968.
34. C.-H.L. Ong, "The Lazy λ-Calculus : an Investigation into the Foundations of Functional Programming", Ph.D. Thesis , University of London, 1988. Also Prize Fellowship Dissertation, Trinity College, Cambridge.
35. C.-H.L. Ong, "The Concurrent Lambda Calculus 1: a General Precongruence Theorem for Applicative Bisimulation", Unpublished Manuscript, Cambridge University Computer Laboratory, 1992.
36. C.-H.L. Ong, "Non-Determinism in a Functional Setting", *LICS '93*, IEEE Computer Soc. Press, Silver Spring, 1993, 275-286.
37. G.D. Plotkin, "LCF Considered as a Programming Language", *Theor. Comp. Sci.* 5, 1977, 223-256.
38. G.D. Plotkin, "A Powerdomain Construction", *SIAM J. of Comp.* 3, 1976, 452-487.
39. G. D. Plotkin, "A Semantics for Static Type Inference", *Info. and Comp.* 109, 1994, 256-299.
40. S.Ronchi della Rocca, B.Venneri, "Principal Type Schemes for an Extended Type Theory", *Theor. Comp. Sci.* 28, 1984, 151-169.
41. D. Sangiorgi, "The Lazy λ-calculus in a Concurrency Scenario", *LICS'92*, IEEE Computer Soc. Press, Silver Spring, 1992, 102-109.
42. D. Scott, "Domains for Denotational Semantics", ICALP'82, *LNCS* 140, Springer-Verlag, Berlin, 1982, 577-613.
43. M.B. Smyth, "Power Domains", *J. Comp. Sys. Sci.* 16, 1978, 23-36.

44. B. Thomsen, " Calculi for Higher-Order Communicating Systems", Ph.D. Thesis, Imperial College, 1990.

45. 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, 488-521.