



Lambda-Calculus, Multiplicities and the pi-Calculus

G rard Boudol, Cosimo Laneve

► **To cite this version:**

G rard Boudol, Cosimo Laneve. Lambda-Calculus, Multiplicities and the pi-Calculus. [Research Report] RR-2581, INRIA. 1995, pp.33. <inria-00074103>

HAL Id: inria-00074103

<https://hal.inria.fr/inria-00074103>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin e au d p t et   la diffusion de documents scientifiques de niveau recherche, publi s ou non,  manant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv s.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

λ -Calculus, Multiplicities and the π -Calculus

G rard Boudol and Cosimo Laneve

N  2581

Juin 1995

PROGRAMME 2



*Rapport
de recherche*



λ -Calculus, Multiplicities and the π -Calculus *

G rard Boudol and Cosimo Laneve**

Programme 2 — Calcul symbolique, programmation et g nie logiciel
Projet meije

Rapport de recherche n   2581 — Juin 1995 — 33 pages

Abstract: In this paper we study the semantics of the λ -calculus induced by Milner's encoding into the π -calculus. We show that the resulting may testing preorder on λ -terms coincides with the inclusion of L vy-Longo trees. To establish this result, we use a refinement of the λ -calculus where the argument of a function may be of limited availability. In our λ -calculus with multiplicities, evaluation is deterministic, but it may deadlock, due to the lack of resources. We show that this feature is enough to make the λ -calculus as discriminating as the π -calculus.

Key-words: functional and concurrent languages, semantics, lambda-calculus

(R sum  : tsvp)

*Partially supported by the ESPRIT Basic Research Project 6454 - CONFER.

**Present address: Universit  di Bologna, Dipartimento di Matematica, Piazza di Porta San Donato 5, 40127 Bologna, Italy.

λ -Calcul, Multiplicités et π -Calcul

Résumé : Le codage du λ -calcul dans le π -calcul donné par Milner induit sur le premier une sémantique que nous étudions dans cet article. Nous montrons que le préordre de test résultant sur les λ -termes coïncide avec l'inclusion des arbres de Lévy-Longo. Pour établir ce résultat, nous utilisons un raffinement du λ -calcul où l'usage de l'argument d'une fonction peut être limité. Dans notre λ -calcul avec multiplicités, l'évaluation est déterministe, mais peut conduire à des blocages, dûs au manque de ressource. Nous montrons que cette propriété confère au calcul un pouvoir de séparation aussi grand que celui du π -calcul.

Mots-clé : langages fonctionnels et parallèles, sémantique, lambda-calcul

1 Introduction

A few years ago, Milner, Parrow and Walker introduced a calculus of mobile processes [20], now called the π -calculus, which is a name passing extension of Milner's CCS [17]. A version of this calculus was first discovered by Engberg and Nielsen [10], following early discussions between Nielsen and Milner (see the conclusion of [20], Part I). The paper [20] nicely demonstrated the usefulness of the name passing discipline, by a series of examples showing clearly its expressiveness. Milner quickly realized that one can encode the λ -calculus into the π -calculus [18], a remarkable achievement, showing the universality of the π -calculus in some sense. This result was further studied and extended, most notably by Sangiorgi [23], who established that the π -calculus is powerful enough to encode a very general form of agent passing.

Milner actually gave two encodings in [18], one for the *lazy* λ -calculus of Abramsky and Ong [3], and another for the weak call-by-value λ -calculus. It turns out that, in both cases, the π -calculus is strictly more expressive than the λ -calculus, in the sense that some λ -terms that are equated using λ -calculus means can be distinguished in the π -calculus. Milner then raised the question: “*Exactly what is the semantics induced upon λ -terms by encoding them into the π -calculus?*” The purpose of this paper is to investigate this question, for the “lazy encoding”.

To introduce our contribution, let us first formulate Milner's question more precisely. In CCS-like calculi, there is a natural notion of *immediate observability*: a process P is immediately observable, in notation $P\Downarrow$, if P exhibits, without any internal computation, a capability of communicating with its environment. That is, P is able to perform an action (input or output). Now a process P is *observable*, in notation $P\Downarrow$, if it may become immediately observable, possibly after some internal computations, that is $\exists P', P \xrightarrow{*} P' \ \& \ P'\Downarrow$. The semantic preorder on the π -calculus that Milner considered in [18] is defined as follows:

$$P \sqsubseteq Q \Leftrightarrow \text{for all contexts } C, C[P]\Downarrow \Rightarrow C[Q]\Downarrow$$

This is known as the *may testing* semantics, also used when dealing with the denotational semantics of the λ -calculus, see [26] for instance. Now, denoting by $\mathcal{E}[[M]]u$ the encoding of a λ -term M in the π -calculus (this involves a name parameter u , the actual value of which is immaterial), the question is to determine exactly what is the preorder on λ -terms given by

$$M \sqsubseteq_{\pi} N \Leftrightarrow \mathcal{E}[[M]]u \sqsubseteq \mathcal{E}[[N]]u$$

Sangiorgi studied a similar question, where the may testing \sqsubseteq is replaced by *weak bisimulation*, denoted here \approx . He showed – see [23, 24], and especially [25], from which we shall borrow some results – that this “ π -semantics” $M \approx_{\pi} N$ on λ -terms coincides with equality in a semantics given by Lévy in [13] for the λ -calculus. In [15], Longo gave a suggestive presentation of Lévy's interpretation, by means of what are now called *Lévy-Longo trees*, a refinement of the well-known Böhm trees (see [4]), suited for the weak λ -calculus where any divergent term as $\Omega = (\lambda xxx)(\lambda xxx)$ is different from $\lambda x \Omega$. Lévy's interpretation provides a rather intensional semantics for λ -terms, which may be regarded as the finest

reasonable semantics one can imagine. This was established by Longo in [15], who showed that any “reasonable” λ -model induces a semantics which is weaker than L evy’s one. Then the π -semantics on λ -terms is at the same time very sharp and still “reasonable”.

In [25], Sangiorgi indicated that one may suspect that may testing equivalence, that is $M \simeq_\pi N$ if and only if $M \sqsubseteq_\pi N$ and $N \sqsubseteq_\pi M$, and bisimulation $M \approx_\pi N$ coincide – a reason should be that λ -terms are “deterministic processes”. This is indeed a consequence of a result of this paper: we show, thus answering Milner’s question, that $M \sqsubseteq_\pi N$ coincides with $M \sqsubseteq_{\mathcal{L}} N$, the preorder induced by L evy’s interpretation, that is the inclusion of L evy-Longo trees. Considering Sangiorgi’s work, the fact that π -calculus semantics of λ -terms is again related to L evy’s semantics should not be too surprising. In particular, the fact that L evy’s semantics is stronger than π -may testing is a direct consequence of results of [25]. However, the proof method we use for showing

$$M \sqsubseteq_\pi N \Rightarrow M \sqsubseteq_{\mathcal{L}} N$$

is completely different from that of [25] for the same implication, stated in terms of bisimulation. In fact, the way we prove it is the main contribution of this paper.

To prove this implication, we use a refinement of the λ -calculus introduced in [6], namely the λ -calculus with multiplicities. To motivate this refinement, let us point out one typical reason why the π -calculus is more discriminating than the λ -calculus. Milner’s encoding translates β -reduction as follows:

$$\mathcal{E}[(\lambda x M)N]u \rightarrow (\nu x)(\mathcal{E}[M]u \mid !x(w)\mathcal{E}[N]w) \approx \mathcal{E}[[N/x]M]u$$

where $!P$ represents an infinite parallel composition ($P \mid \dots \mid P \dots$). This is needed to adequately model the fact that, in an application RN , the argument N is *infinitely available* for the function R . Indeed, if $R = \lambda x M$, this term reduces to $[N/x]M$ where the argument is copied within M as many times as there are free occurrences of x . One cannot predict the “multiplicity” of x in M , because M could be reduced to another term where this variable is duplicated. This is the case for instance if $M = (\mathbf{2}x)$ where $\mathbf{2} = \lambda f y. f(fy)$.

Now, in the π -calculus, one is not compelled to use infinitely available resources. For instance, one could write something like a λ -term where the argument is available at most m times, namely:

$$(\nu x)(\mathcal{E}[M]u \mid \underbrace{x(w)\mathcal{E}[N]w \mid \dots \mid x(w)\mathcal{E}[N]w}_m)$$

Our purpose is to show that the limited availability of a resource is precisely the reason why the π -calculus is more discriminating than the λ -calculus. Firstly, we note that this provides us with extra discriminating power. For instance, the two terms xx and $x(\lambda y. xy)$ are equated in the canonical denotational semantics of the lazy λ -calculus (see [3]), while, in the π -calculus, one may distinguish them by providing just one sample of the identity $\mathbf{I} = \lambda z z$ for x . More precisely, the term

$$(\nu x)(\mathcal{E}[xx]u \mid x(w)\mathcal{E}[\mathbf{I}]w)$$

is not observable – it reduces to a deadlock –, whereas

$$(\nu x)(\mathcal{E}[x(\lambda y.xy)]u \mid x(w)\mathcal{E}[\mathbf{I}]w)$$

converges.

This led us to introduce a λ -calculus with explicit, finite or infinite *multiplicities*, writing MN^m where $m \in \mathbb{N} \cup \{\infty\}$ to mean that N is available at most m times for M . As a particular case, we get the usual λ -terms, where all the multiplicities are infinite – in which case we may omit them, to keep the standard notation. We call this new calculus the λ_m -calculus. Obviously, as in the π -calculus, evaluating a λ_m -term may end up with a *deadlock*. This is the case of $(\lambda x.xx)\mathbf{I}^1$, for instance.

Actually, the π -calculus suggests an even more liberal extension of the λ -calculus. After all, one is not compelled to use copies of the same argument as resources. One could write in the π -calculus:

$$(\nu x)(\mathcal{E}[M]u \mid x(w)\mathcal{E}[N_1]w \mid \cdots \mid x(w)\mathcal{E}[N_m]w)$$

For example, it is possible to encode in this way a non-deterministic *internal choice*:

$$\mathcal{E}[M \oplus N]u = (\nu x)(\mathcal{E}[x]u \mid x(w)\mathcal{E}[M]w \mid x(w)\mathcal{E}[N]w)$$

The further extension of the λ -calculus where the arguments, that is P in MP are *bags* of terms was introduced in [6]. The syntax for bags follows that of the π -calculus: $\mathbf{0}$ is the empty bag and $(P \mid Q)$ is the (multiset) union of P and Q . However, we write M^∞ for a bag consisting of M with an infinite multiplicity, rather than $!M$. We call this the λ -calculus *with resources*, or λ_r -calculus. Obviously, it contains the λ -calculus with multiplicities, since one may let $N^m = (N \mid \cdots \mid N)$, m times.

The λ_m and λ_r -calculi are similar in spirit to the λ -calculus. They are all based on the idea of a function applied to arguments, or more generally to bags of arguments. However, the evaluation mechanisms exhibit different features. In the lazy λ -calculus, the evaluation is deterministic, that is, at most one reduction is possible at each step, and may either diverge or terminate on a value, i.e. an abstraction. In the λ_m -calculus, evaluation is still deterministic, but in addition it may deadlock. The λ_r -calculus further adds possible *non determinism* in the evaluation, because one may choose any of the resources that a bag contains to instantiate a variable. Therefore, it is natural to refine not only the syntax and evaluation, but also the semantics.

In this paper we take the view that divergence, deadlock and convergence are distinguished by the observer. More precisely, divergence is not observed, while deadlock and convergence are observed differently. Then the may testing semantics for λ_m and λ_r is defined as follows:

M is observationally less than N if and only if for all contexts C , if $C[M]$ may deadlock then $C[N]$ may deadlock, and if $C[M]$ may converge to a value, then $C[N]$ may converge to a value.

By taking the contexts to be λ_m -contexts or λ_r -contexts we get two preorders \sqsubseteq_m and \sqsubseteq_r , the second one being obviously stronger. Our main result is that, as far as λ -terms are concerned, explicit multiplicities – that is, potential deadlocks – are enough to distinguish as much as the π -calculus and L evy’s interpretation. In particular, non-deterministic features like internal choice are not needed for testing λ -terms. That is, for $M, N \in \Lambda$:

$$M \sqsubseteq_{\mathcal{L}} N \Leftrightarrow M \sqsubseteq_r N \Leftrightarrow M \sqsubseteq_m N \Leftrightarrow M \sqsubseteq_{\pi} N$$

We should also mention that, if we are interested in equivalences, not preorders, then a weaker observational semantics, where deadlocks are not observed, is enough to ensure a similar result, see [8, 9]. These results may be surprising, because the λ -calculus with multiplicities is purely deterministic, with no parallel facility. This contrasts with a previous result by Sangiorgi showing that “*non-determinism is exactly what is necessary to add to the λ -calculus to make it as discriminating as the π -calculus*” – in fact, the unary non-deterministic operator given by $\uplus M = (M \oplus \Omega)$ is enough, see [24]. Remember, however, that Sangiorgi uses a bisimulation semantics, for which non-determinism has some flavour of introducing potential deadlocks. For instance $\uplus M \rightarrow \Omega$ means that M , regarded as a resource, may vanish. Here we may have a different conclusion, regarding Milner’s question: *the possibility of deadlocks is essentially what the π -calculus adds to the lazy λ -calculus*.

The paper is organized as follows: the first two sections give a brief account of the π -calculus and the weak λ -calculus. Then we introduce the λ -calculus with resources and the λ -calculus with multiplicities, their syntax and evaluation mechanism. Next we define the observational semantics and show the context lemma. The last two sections are devoted to the proof of our main result: we first show, establishing the approximation lemma and the separation lemma, that L evy’s semantics coincides with that induced by the λ_r and λ_m -calculi. Then we show that these are weaker than the π -calculus semantics. The converse is proved using Sangiorgi’s results in [25]. Finally we conclude by comparing our results with related work.

2 The π -calculus

In this paper we adopt the same, asynchronous “mini” π -calculus as in [25], built upon a given denumerable set \mathcal{N} of *names*, ranged over by $u, v, w \dots$. The differences with other versions, e.g. in [20, 18, 19], are the following: firstly, we do not use the sum and matching constructs $P + Q$ and $[u = v]P$. The *input* construct $u(v_1, \dots, v_k)P$ allows P to receive on the channel u several names simultaneously, as in the polyadic π -calculus of [19]. Correspondingly, an output sends several names simultaneously. However, it is not a guard. That is, it takes the form of an asynchronous *message* $\bar{u}v_1 \dots v_k$. Besides the “empty” process 0 , the remaining constructs are *parallel composition* $(P \mid Q)$, *replication* $!P$ and *restriction*, or scoping $(\nu u)P$. The syntax of this (mini) π -calculus is thus:

$$P ::= 0 \mid \bar{u}v_1 \dots v_k \mid u(v_1, \dots, v_k)P \mid (P \mid P) \mid !P \mid (\nu u)P$$

where $k \geq 0$ and u, v_1, \dots, v_k are names. We let Π denote the set of π -terms, ranged over by P, Q, \dots . The names v_1, \dots, v_k in an input prefix $u(v_1, \dots, v_k)$ are assumed to be pairwise distinct. They are bound by this construct, and similarly u is bound by a restriction (νu) . We denote by $\text{fn}(P)$ and $\text{bn}(P)$ the set of names occurring respectively free and bound in P , and $\text{nm}(P)$ denotes the union of these two sets. We denote by $[u/v]P$ the operation of substituting the name u for v in P . This may require some renaming, to avoid binding u , as in the λ -calculus. More generally, we denote by $[u_1, \dots, u_n/v_1, \dots, v_n]P$ the simultaneous substitution of u_1, \dots, u_n for v_1, \dots, v_n respectively in P . We denote by $P =_\alpha Q$ the α -conversion, that is the least congruence satisfying

$$u(v_1, \dots, v_k)P =_\alpha u(w_1, \dots, w_k)[w_1/v_1] \cdots [w_k/v_k]P$$

where w_1, \dots, w_k are distinct names not in $\text{nm}(P)$, and

$$(\nu u)P =_\alpha (\nu v)[v/u]P$$

where $v \notin \text{nm}(P)$. We shall sometimes write $\bar{u}\tilde{v}$ for $\bar{u}v_1 \cdots v_k$, and similarly for the input prefix $u(\tilde{v})P$, for the substitution $[\tilde{w}/\tilde{v}]P$, and for a sequence of restrictions $(\nu\tilde{u})P$.

The *structural equivalence* $P \equiv Q$ over π -terms is the least congruence containing $=_\alpha$, and satisfying the following equations:

$$\begin{aligned} (P \mid (Q \mid R)) &\equiv ((P \mid Q) \mid R) & ((\nu u)P \mid Q) &\equiv (\nu u)(P \mid Q) \quad (u \notin \text{fn}(Q)) \\ (P \mid Q) &\equiv (Q \mid P) & (\nu u)(\nu v)P &\equiv (\nu v)(\nu u)P \\ (P \mid 0) &\equiv P & (\nu u)0 &\equiv 0 \\ !P &\equiv (P \mid !P) \end{aligned}$$

The *reduction* relation $P \rightarrow P'$ is given by the following rules:

$\frac{(\bar{u}w_1 \cdots w_k \mid u(v_1, \dots, v_k)P) \rightarrow [w_1, \dots, w_k/v_1, \dots, v_k]P \quad Q \equiv P, P \rightarrow P'}{Q \rightarrow P'}$
$\frac{P \rightarrow P'}{(P \mid Q) \rightarrow (P' \mid Q)} \quad \frac{P \rightarrow P'}{(\nu u)P \rightarrow (\nu u)P'}$

We may simply write the basic reduction step as

$$(\bar{u}\tilde{w} \mid u(\tilde{v})P) \rightarrow [\tilde{w}/\tilde{v}]P$$

provided that the sequences of names \tilde{w} and \tilde{v} have the same length. To distinguish π -reduction from other notions of reduction we shall consider, we sometimes use the notation $P \rightarrow_\pi P'$. The reflexive and transitive closure of this relation is denoted $\xrightarrow{*}$, as usual. The predicate $P \downarrow$ of *immediate observability* is defined by

$$P \downarrow \Leftrightarrow \exists u. P \downarrow u$$

where $P \downarrow u$ is the least relation satisfying

$$\begin{aligned} \bar{u}v_1 \cdots v_k \downarrow u \quad \text{and} \quad u(v_1, \dots, v_k)P \downarrow u \\ P \downarrow u \Rightarrow (P \mid Q) \downarrow u \quad \& \quad !P \downarrow u \\ P \downarrow u \quad \& \quad u \neq v \Rightarrow (\nu v)P \downarrow u \end{aligned}$$

The following is easy to check:

Remark 2.1 $P \downarrow u$ if and only if either

- (i) $P \equiv (\nu \tilde{w})(\bar{u}\tilde{v} \mid Q)$ with u not in \tilde{w} , or
- (ii) $P \equiv (\nu \tilde{w})(u(\tilde{v})R \mid Q)$ with u not in \tilde{w} .

The predicate $P \Downarrow$ of *observability* is given by

$$P \Downarrow \Leftrightarrow \exists P'. P \xrightarrow{*} P' \quad \& \quad P' \downarrow$$

Like for reduction, we shall sometimes use the notations $P \downarrow_\pi$ and $P \Downarrow_\pi$. A context, or more accurately a π -context is a term C written using the syntax of the π -calculus enriched with a constant $[\]$, the hole. We denote by $C[P]$ the term obtained by filling the hole with P in C . Note that some free names of P may be bound by the context.

Definition 2.2 (Observational Semantics) *The observational semantics of the π -calculus is the preorder defined as follows:*

$$P \sqsubseteq_\pi Q \Leftrightarrow \text{for all } \pi\text{-contexts } C, C[P] \Downarrow \Rightarrow C[Q] \Downarrow$$

This preorder is clearly a precongruence, that is

$$P \sqsubseteq_\pi Q \Rightarrow C[P] \sqsubseteq_\pi C[Q]$$

We denote by \simeq_π the equivalence associated with the observational semantics, that is

$$P \simeq_\pi Q \Leftrightarrow P \sqsubseteq_\pi Q \quad \& \quad Q \sqsubseteq_\pi P$$

For instance, we have $0 \simeq_\pi (\nu u)\bar{u}$, therefore 0 is not a primitive of the calculus. However, it is convenient to include this constant, and more specifically to have the law $(P \mid 0) \equiv P$.

In the following we shall use a result which is proved by Sangiorgi in [25]. He defines an equivalence on the π -calculus, called “weak ground bisimilarity”, based on labelled transitions between π -terms. These transitions may be defined as follows: the labels are either input actions, that is $a = u(v_1, \dots, v_k)$, or output actions $a = (\nu w_1) \cdots (\nu w_n)\bar{u}v_1 \cdots v_k$, and:

- (i) $P \xrightarrow{a} P'$ where $a = u(v_1, \dots, v_k)$ if and only if $P \equiv (\nu u_1) \cdots (\nu u_n)(u(v_1, \dots, v_k)R \mid Q)$ with $u \notin \{u_1, \dots, u_n\}$ and $v_i \notin \text{fn}(Q)$, and $P' = (\nu u_1) \cdots (\nu u_n)(R \mid Q)$

(ii) $P \xrightarrow{\alpha} P'$ where $a = (\nu w_1) \cdots (\nu w_m) \bar{u}v_1 \cdots v_k$ if and only if $P \equiv (\nu u_1) \cdots (\nu u_n)(\bar{u}v_1 \cdots v_k \mid Q)$ with $u \notin \{u_1, \dots, u_n\}$, $\{w_1, \dots, w_m\} = \{u_1, \dots, u_n\} \cap \{v_1, \dots, v_k\}$ and $P' = (\nu s_1) \cdots (\nu s_h)Q$ where $\{s_1, \dots, s_h\} = \{u_1, \dots, u_n\} - \{w_1, \dots, w_m\}$.

Clearly $P \Downarrow$ if and only if $P \xrightarrow{\alpha} P'$ for some a and P' . Now by just removing the *symmetry* requirement from Sangiorgi's Definition 2.2, one may define:

Definition 2.3 (Simulation) *A relation \mathcal{S} on π -terms is a simulation if it satisfies:*

- (i) *if PSQ and $P \rightarrow P'$ then there exists Q' such that $Q \xrightarrow{*} Q'$ and $P' S Q'$,*
 - (ii) *if PSQ and $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{*} \xrightarrow{\alpha} \xrightarrow{*} Q'$ and $P' S Q'$.*
- P simulates Q , written $P \preceq_{\pi} Q$, if PSQ for some simulation \mathcal{S} .*

Then we have:

Lemma 2.4 *The relation \preceq_{π} is a precongruence.*

This is essentially Sangiorgi's Corollary 3.6 [25], once one remarks that symmetry does not play any rôle in the proof.

Corollary 2.5 $P \preceq_{\pi} Q \Rightarrow P \sqsubseteq_{\pi} Q$

This is obvious since

$$P \Downarrow \ \& \ P \preceq_{\pi} Q \Rightarrow Q \Downarrow$$

3 The weak λ -calculus

The denumerable set \mathcal{X} of variables used to build λ -terms is assumed to be a subset of \mathcal{N} , such that $\mathcal{N} - \mathcal{X}$ is infinite. When we write $u, v, w \dots$ we usually mean names that are not variables, and we use $x, y, z \dots$ to range over \mathcal{X} . We recall that λ -terms are given by the following grammar:

$$M ::= x \mid \lambda x M \mid (MM)$$

We assume the reader is familiar with the notions of free and bound variables, α -conversion $M =_{\alpha} N$, substitution $[N/x]M$, and β -conversion $M =_{\beta} N$, that is the congruence generated by $(\lambda x M)N = [N/x]M$, see [4]. We use the standard notations: Λ is the set of λ -terms, Λ° is the subset of closed terms, $\lambda x_1 \dots x_n. M$ stands for $\lambda x_1 \dots \lambda x_n M$, and $MN_1 \cdots N_n$ abbreviates $(\cdots (MN_1) \cdots N_n)$.

In the λ -calculus literature, the word *weak* refers to the failure or disregarding of the ξ -rule of β -conversion, that is $M = N \Rightarrow \lambda x M = \lambda x N$ (see Howard's weak λ -equality in [11]). In particular, if ξ is not considered as a computation rule, then a closed normal form is a "weak head normal form", that is simply an abstraction $\lambda x M$ – this is the notion of value adopted in the operational semantics of programming languages, see [22] for instance. From the point of view of observational semantics, this amounts to identifying all the strongly

unsolvable terms, but not necessarily the unsolvable ones. Typically, one has $\lambda x\Omega \neq \Omega$ in the weak λ -calculus, while these two terms must be equated in *sensible* λ -theories (see [4]).

The weak λ -calculus was studied in depth by Abramsky and Ong (see [2, 3]), who call it the *lazy* λ -calculus. One may observe that, for M closed, $M =_\beta \lambda xN$ if and only if M has a normal form with respect to the “lazy” reduction relation over λ -terms, given by the following two rules:

$$\boxed{(\lambda xM)N \rightarrow [N/x]M \quad \frac{M \rightarrow M'}{MN \rightarrow M'N}}$$

Then, defining the observability predicate by

$$M \downarrow_\ell \Leftrightarrow \exists M'. M \xrightarrow{*}_\ell M' \ \& \ M' \downarrow_\ell$$

where $M \downarrow_\ell$ simply means that M is an abstraction λxN , one defines the (*weak*, or *lazy*) *observational semantics* for the λ -calculus exactly as we did for the π -calculus:

$$M \sqsubseteq_\ell N \Leftrightarrow \text{for all contexts } C, C[M] \downarrow_\ell \Rightarrow C[N] \downarrow_\ell$$

This is the semantics introduced by Abramsky, who called it “applicative bisimulation”, see [2, 3]. One should note that, although ξ is not taken as a computation rule, it is semantically valid. More generally, one has:

$$M =_\beta N \Rightarrow M \simeq_\ell N$$

In [13], L evy defined an interpretation of the weak λ -calculus, based on a refinement of Wadsworth’s notion of syntactic approximant [26], as follows:

Definition 3.1 (L evy’s Interpretation) *The set \mathcal{L} of (lazy) approximants, ranged over by A, B, \dots , is the least subset of Λ containing $\lambda x_1 \dots x_n. \Omega$, and $\lambda x_1 \dots x_n. xA_1 \dots A_m$ whenever $A_i \in \mathcal{L}$. For $M \in \Lambda$, the direct approximation of M is the term $\varpi(M)$ of \mathcal{L} inductively defined by:*

$$\begin{aligned} \varpi(\lambda x_1 \dots x_n. (\lambda y. M)NM_1 \dots M_k) &= \lambda x_1 \dots x_n. \Omega \\ \varpi(\lambda x_1 \dots x_n. yM_1 \dots M_k) &= \lambda x_1 \dots x_n. y \varpi(M_1) \dots \varpi(M_k) \end{aligned}$$

The interpretation of $M \in \Lambda$ is $\mathcal{A}(M) = \{\varpi(N) \mid M =_\beta N\}$. L evy’s preorder on λ -terms, denoted $M \sqsubseteq_{\mathcal{L}} N$, is the inclusion of sets of approximants $\mathcal{A}(M) \subseteq \mathcal{A}(N)$. The equality $M =_{\mathcal{L}} N$ is $\mathcal{A}(M) = \mathcal{A}(N)$.

L evy’s main result is that his preorder is a precongruence. An immediate consequence is

$$M \sqsubseteq_{\mathcal{L}} N \Rightarrow M \sqsubseteq_\ell N$$

One may characterize L evy’s preorder on approximants: $A \sqsubseteq_{\mathcal{L}} B$ if and only if A is a *prefix* of B , where the prefix ordering is the precongruence \preceq on approximants generated by $\Omega \preceq A$. The Church-Rosser property has the following consequence (see [13]):

Lemma 3.2 *For any $M \in \Lambda$, the set $\mathcal{A}(M)$ is directed with respect to the prefix preorder, namely*

$$\forall A', A'' \in \mathcal{A}(M) \exists A \in \mathcal{A}(M). A' \preceq A \ \& \ A'' \preceq A$$

Moreover, it is easy to see that $\mathcal{A}(M)$ is in fact an *ideal*, that is it is downward-closed with respect to the prefix ordering:

$$A \in \mathcal{A}(M) \ \& \ B \preceq A \Rightarrow B \in \mathcal{A}(M)$$

This is because any term M is β -convertible to a redex, namely $(\mathbf{I} M)$ where \mathbf{I} is the identity λxx , whose direct approximation is Ω . In particular, we have $\Omega \in \mathcal{A}(M)$ for any M .

In [15], Longo gave a suggestive presentation of Lévy's interpretation, by means of what is now called Lévy-Longo trees, which were further studied by Ong in [21]. These are refinements of the well-known Böhm trees (see [4]), adapted to the lazy regime. The Lévy-Longo trees are possibly infinite, node-labelled trees, where the labels are either Υ , representing terms of infinite order as $\Xi = (\lambda fx. ff)(\lambda fx. ff)$, or $\lambda x_1 \dots x_n. \perp$, representing terms as $\lambda x_1 \dots x_n. \Omega$, or $\lambda x_1 \dots x_n. x$, representing the "head" of a solvable term, as in Böhm trees. To define these trees, let us first recall the notion of a λ -term of *proper order* n , with $n \in \mathbb{N} \cup \{\infty\}$:

1. $M \in \text{PO}_0 \Leftrightarrow \forall M'. M \xrightarrow{*} M' \Rightarrow \exists M''. M' \rightarrow M''$
2. $M \in \text{PO}_{n+1} \Leftrightarrow \exists x \exists M' \in \text{PO}_n. M \xrightarrow{*} \lambda x M'$
3. $M \in \text{PO}_\infty \Leftrightarrow \forall n \exists x_1, \dots, x_n \exists M'. M =_\beta \lambda x_1 \dots x_n. M'$

The *Lévy-Longo tree* of a λ -term M , $\text{LT}(M)$, is defined inductively as follows:

1. $\text{LT}(M) = \Upsilon$ if $M \in \text{PO}_\infty$;
2. $\text{LT}(M) = \lambda x_1 \dots x_n. \perp$ if $M \in \text{PO}_n$;
3. $\text{LT}(M) =$

$$\begin{array}{c} \lambda x_1 \dots x_n. x \\ \swarrow \quad \searrow \\ \text{LT}(M_1) \quad \dots \quad \text{LT}(M_k) \end{array}$$
if $M =_\beta \lambda x_1 \dots x_n. x M_1 \dots M_k$.

To recover Lévy's ordering $M \sqsubseteq_{\mathcal{L}} N$ on the tree representation, one defines an operation $\lambda x T$ on trees, consisting in prefixing the label of the root of T by λx , with the rule that $\lambda x \Upsilon = \Upsilon$. Then a tree T is less than T' whenever T' is obtained from T by replacing some leaves labelled $\lambda x_1 \dots x_n. \perp$ in T by trees $\lambda x_1 \dots x_n. T''$. An example of infinite Lévy-Longo tree is provided by Wadsworth's combinator \mathbf{J} , satisfying $\mathbf{J} =_\beta \lambda xy. x(\mathbf{J}y)$, which may be

defined by $\mathbf{J} = (\lambda f \lambda x y. x(f f y))(\lambda f \lambda x y. x(f f y))$. The tree for this term is:

$$\begin{array}{lcl} \text{LT}(\mathbf{J}) & = & \lambda x y_0. x \\ & & | \\ & & \lambda y_1. y_0 \\ & & | \\ & & \lambda y_2. y_1 \\ & & \vdots \end{array}$$

In the rest of this section we start relating the λ -calculus with the π -calculus. The main result here, which we prove using Sangiorgi’s results in [25], is that L evy’s interpretation is *adequate* with respect to the “ π -semantics”. Let us first recall the encoding of the λ -calculus into the π -calculus given by Milner [18], written in the “asynchronous” style. It is the mapping

$$\mathcal{E}[\cdot]: \Lambda \times (\mathcal{N} - \mathcal{X}) \rightarrow \Pi$$

defined as follows:

$\begin{aligned} \mathcal{E}[[x]]u &= \bar{x}u \\ \mathcal{E}[[\lambda x M]]u &= u(x, v)\mathcal{E}[[M]]v \\ \mathcal{E}[[MN]]u &= (\nu v)(\mathcal{E}[[M]]v \mid (\nu x)(\bar{v}xu \mid \mathcal{E}[[x := N]])) \quad \text{where } x \notin \text{fv}(N) \text{ and} \\ \mathcal{E}[[x := N]] &= !x(w)\mathcal{E}[[N]]w \end{aligned}$
--

This encoding establishes a close correspondence between the reduction relations of the two calculi, see [18, 25]. For instance we have:

$$\mathcal{E}[[\lambda x M]N]u \rightarrow (\nu x)(\mathcal{E}[[M]]u \mid \mathcal{E}[[x := N]])$$

Note however that $\mathcal{E}[[\lambda x M]N]u$ reduces to something “similar to” the encoding of $[^N/x]M$, though not exactly to $\mathcal{E}[[N/x]M]u$.

We shall abusively denote by $M \sqsubseteq_{\pi} N$ the relation $\mathcal{E}[[M]]u \sqsubseteq_{\pi} \mathcal{E}[[N]]u$, which does not depend on the name u . This is the preorder on λ -terms we are interested in. In [25], Sangiorgi gives a characterization of the equality of λ -terms in L evy’s interpretation. By removing the *symmetry* requirement from his Definition 7.7, one can characterize the preorder $\sqsubseteq_{\mathcal{L}}$ as follows: $M \sqsubseteq_{\mathcal{L}} N$ if and only if

- (i) either M diverges, that is $M \in \text{PO}_0$,
- (ii) or $M \xrightarrow{*}_{\ell} \lambda x M'$, and there exists N' such that $N \xrightarrow{*}_{\ell} \lambda x N'$ and $M' \sqsubseteq_{\mathcal{L}} N'$,
- (iii) or $M \xrightarrow{*}_{\ell} x M_1 \cdots M_n$ and $N \xrightarrow{*}_{\ell} x N_1 \cdots N_n$ for some N_1, \dots, N_n such that $M_i \sqsubseteq_{\mathcal{L}} N_i$.

Then Sangiorgi shows

$$M \sqsubseteq_{\mathcal{L}} N \Rightarrow M \preceq_{\pi} N$$

This is his Theorem 7.15 [25], once one remarks that the symmetry requirement is inessential. Then an immediate consequence of this result and Corollary 2.5 is the following:

Theorem 3.3 (Adequacy of Lévy’s Interpretation)

$$M \sqsubseteq_{\mathcal{L}} N \Rightarrow M \sqsubseteq_{\pi} N$$

It is possible to give a direct proof of this theorem, by showing that for any π -context C , if $C[\mathcal{E}[[M]]u] \Downarrow$ then there is only a finite amount of information about M that needs to be known to ensure the observability of $C[\mathcal{E}[[M]]u]$. That is:

$$C[\mathcal{E}[[M]]u] \Downarrow \Leftrightarrow \exists A \in \mathcal{A}(M) \ C[\mathcal{E}[[A]]u] \Downarrow$$

This may be proved by induction on the length of the reduction of $C[\mathcal{E}[[M]]u]$ into an immediately observable term, and by induction on the context C . However, since a “redex” – that is, a subterm performing a reduction step – is “distributed” in a π -term, we have to analyze the contribution of M to the reduction. This may be done using the labelled transitions of $\mathcal{E}[[M]]u$. Moreover, when $M \rightarrow M'$, the π -term $\mathcal{E}[[M]]u$ does not directly reduce to $\mathcal{E}[[M']]u$, but rather it reduces to a term which is equivalent, in a very strong sense, to $\mathcal{E}[[M']]u$. Therefore, a direct proof of the theorem would not be very different from the proof of Sangiorgi’s result, which relies on a thorough analysis of the labelled transitions of encodings of λ -terms.

4 The λ -calculus with resources

As we said in the introduction, to prove the converse of Theorem 3.3 we shall use a refinement of the λ -calculus introduced in [6], where the argument R in an application (MR) is a bag of terms. To define the evaluation in this calculus, it is convenient to use the notion of *explicit substitution* of Curien et al. [1], written $M\langle R/x \rangle$. Then the syntax of our λ -calculus with resources, or λ_r -calculus, is as follows:

$$\begin{aligned} E & ::= x \mid \lambda x E \mid (ER) \mid E\langle R/x \rangle \\ R & ::= 0 \mid E \mid (R \mid R) \mid E^\infty \end{aligned}$$

To avoid any confusion with usual λ -terms, denoted M, N, \dots we use E, F, \dots to range over terms of our calculus. The set of terms is Λ_r . The bags of terms will be denoted P, Q, \dots when no confusion with π -terms may arise.

We denote by $\text{fv}(E)$ and $\text{bv}(E)$ the sets of free and bound variables of E . This is defined as usual, except that x is bound in E by the construct $\langle R/x \rangle$, that is

$$\text{fv}(E\langle R/x \rangle) = (\text{fv}(E) - \{x\}) \cup \text{fv}(R)$$

where $\text{fv}(R)$ is defined in the obvious way: $\text{fv}(P \mid Q) = \text{fv}(P) \cup \text{fv}(Q)$, and so on. The set of closed terms is denoted Λ_r° . We shall consider λ_r -terms up to α -conversion. This is defined as usual, with the additional clause that $([z/x]E)\langle R/z \rangle =_\alpha E\langle R/x \rangle$, where z is neither free nor bound in E . The *structural equivalence* over Λ_r is the least congruence containing α -conversion and satisfying:

$$\begin{aligned}
(P \mid (Q \mid R)) &\equiv ((P \mid Q) \mid R) & EP\langle Q/x \rangle &\equiv E\langle Q/x \rangle P \quad (*) \\
(P \mid Q) &\equiv (Q \mid P) & E\langle P/x \rangle \langle Q/z \rangle &\equiv E\langle Q/z \rangle \langle P/x \rangle \quad (**) \\
(P \mid 0) &\equiv P \\
E^\infty &\equiv (E \mid E^\infty)
\end{aligned}$$

(*) where $x \notin \text{fv}(P)$

(**) where $x \neq z$, $x \notin \text{fv}(Q)$ and $z \notin \text{fv}(P)$.

Note that the substitution items may always be pushed on the right of a term. That is, $E\langle Q/x \rangle P$ may always be regarded, by possibly renaming x , as $EP\langle Q/x \rangle$. Therefore, any term E is, up to structural equivalence, of the form $HP_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$ where H is either a variable or an abstraction.

The reduction relation on Λ_r is split into two parts: one is the usual (weak) β -reduction, written with explicit substitutions. The second part deals with the management of substitution. Since in our calculus, the resources may be of limited availability, we will not “perform” the substitutions $\langle P/x \rangle$ by distributing them over the subterms. We will rather use them in a delayed manner, waiting for a resource to be actually needed for x . Assuming that any term E has the shape $HP_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$ where $H = \lambda xF$ or $H = x$, we may describe intuitively the reduction relation as follows:

(1) if H is an abstraction λxF , there are two cases: either $n = 0$, in which case E is a normal form (a *closure*, that is an abstraction within the environment $\langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$), or there is an argument, i.e. $n \neq 0$, in which case the following reduction takes place:

$$E \rightarrow F\langle P_1/x \rangle P_2 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$$

This is formalized by saying that the reduction rules include the β -rule, written with explicit substitutions, and allow to perform computations in the left subterm of ER and $E\langle R/x \rangle$. Moreover, one should be able to first transform E into a term having the appropriate shape. Then our first rules for reduction are:

$(\lambda xE)R \rightarrow E\langle R/x \rangle$	$\frac{F \equiv E, E \rightarrow E'}{F \rightarrow E'}$
$\frac{E \rightarrow E'}{ER \rightarrow E'R}$	$\frac{E \rightarrow E'}{E\langle R/x \rangle \rightarrow E'\langle R/x \rangle}$

(2) if H is a variable x (the *head* variable), one looks for the first substitution for it in the environment $\langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$. Assume that i is the first index such that $x_i = x$. Then

one fetches for x a resource out of Q_i , that is a term $F \in \Lambda_r$ such that $Q_i \equiv (F \mid R)$. The rest R is left for future use. Roughly, the following reduction takes place in this case:

$$E \rightarrow FP_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle R/x_i \rangle \cdots \langle Q_k/x_k \rangle$$

One should be careful not to bind any free variable of F . That is, one should first rename the variables x_1, \dots, x_i , so that they do not occur in F . To state the “fetch” rule, which is the last rule for reduction, it is convenient to introduce an auxiliary relation $E \langle F/x \rangle \rightsquigarrow E'$, meaning that x is the head variable of E , and that E' is obtained by placing F in the head position in E . The rules are as follows:

$\frac{E \langle F/x \rangle \rightsquigarrow E'}{E \langle (F \mid R)/x \rangle \rightsquigarrow E' \langle R/x \rangle} \quad (*)$	$x \langle E/x \rangle \rightsquigarrow E$
$\frac{E \langle F/x \rangle \rightsquigarrow E'}{ER \langle F/x \rangle \rightsquigarrow E'R}$	$\frac{E \langle F/x \rangle \rightsquigarrow E'}{E \langle R/z \rangle \langle F/x \rangle \rightsquigarrow E' \langle R/z \rangle} \quad (**)$

(*) the *fetch* rule, where $x \notin \text{fv}(F)$.

(**) where $x \neq z$ and $z \notin \text{fv}(F)$.

To distinguish the λ_r -reduction from the one of the π and λ -calculi, we shall use the notation $E \rightarrow_r F$. It is not difficult to check that the rules for reduction appropriately formalize the computation steps we intuitively described above, that is:

Lemma 4.1 $E \rightarrow_r E'$ if and only if

(i) either $E \equiv (\lambda x F)P_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle$ with $n > 0$, and

$$E' = F \langle P_1/x \rangle P_2 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle,$$

(ii) or $E \equiv x_i P_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle F \mid R/x_i \rangle \cdots \langle Q_k/x_k \rangle$ where $j \leq i \Rightarrow x_j \notin \text{fv}(F)$

$$\text{and } j < i \Rightarrow x_j \neq x_i, \text{ and } E' = FP_1 \cdots P_n \langle Q_1/x_1 \rangle \cdots \langle R/x_i \rangle \cdots \langle Q_k/x_k \rangle.$$

This reduction process is more like evaluation in an abstract machine setting, where the states have the form $(E, P_1 \cdots P_n, \langle Q_1/x_1 \rangle \cdots \langle Q_k/x_k \rangle)$, than a preorder associated with an equational theory, as β -reduction, possibly with explicit substitutions, is usually presented (see [1, 4] for instance). It is possible to define a notion of reduction in this broader sense for the λ_r -calculus, but this is not relevant for the purpose of this paper. One may immediately observe that λ_r -reduction exhibits some features that make it very different from β -reduction, namely:

(1) *non-determinism*. Since the composition $(P \mid Q)$ of bags of resources is commutative, the fetch rule is non deterministic: one may fetch any of the resources that a bag contains. This is best exemplified by defining a non deterministic *internal choice*:

$$(E \oplus F) =_{\text{def}} x \langle E \mid F/x \rangle \quad x \notin \text{fv}(E) \cup \text{fv}(F)$$

Clearly

$$(E \oplus F) \rightarrow_r E\langle F/x \rangle \quad \text{and} \quad (E \oplus F) \rightarrow_r F\langle E/x \rangle$$

Now any reasonable semantics should equate $E\langle P/x \rangle$ and E whenever $x \notin \text{fv}(E)$, therefore $(E \oplus F)$ adequately represents the choice between E and F .

(2) *deadlocks*. A deadlock arises whenever a resource is both needed and absent. Typically, a term like $x P_1 \cdots P_n \langle 0/x \rangle$ cannot perform any reduction. Therefore, besides the usual normal forms – abstractions calling for an argument –, there is a new kind of irreducible term: variables waiting in vain for a resource.

One may identify which construct of Λ_r is responsible for a particular feature of the evaluation process. In other words, one may define various sub-calculi of Λ_r where evaluation is constrained in some way. Clearly, non-determinism comes from the fact that a bag may contain two different terms. Then a natural restriction to consider is to deal with bags made of copies of the same term. Such a bag is just a term with an explicit finite or infinite *multiplicity*, defined as follows:

$$\begin{aligned} E^0 &= 0 \\ E^{m+1} &= (E \mid E^m) \end{aligned}$$

The λ -calculus with multiplicities is the sub-calculus Λ_m of Λ_r given by the grammar:

$$E ::= x \mid \lambda x E \mid (E E^m) \mid E\langle E^m/x \rangle$$

where $m \in \mathbb{N} \cup \{\infty\}$. This is the calculus we considered in [8, 9]. In particular, we showed in [8] that this sub-calculus is *deterministic* in the sense that, up to structural equivalence, a λ_m -term may perform at most one reduction in one step:

Lemma 4.2 *For any $E, F \in \Lambda_m$, if $E \equiv F$ then*

$$E \rightarrow_r E' \ \& \ F \rightarrow_r F' \ \Rightarrow \ E' \equiv F'$$

As a matter of fact, the reduction relation in Λ_m may be described in a slightly more precise way than as the restriction of the λ_r -reduction to λ_m -terms. Indeed, one can see that the laws for structural equivalence regarding the bags are useless when considering λ_m -terms. Namely, the structural equivalence \equiv_m on Λ_m is the least congruence containing α -conversion and satisfying

$$E G^n \langle F^m/x \rangle \equiv_m E \langle F^m/x \rangle G^n \quad \text{and} \quad E \langle G^n/x \rangle \langle F^m/z \rangle \equiv_m E \langle F^m/z \rangle \langle G^n/x \rangle$$

with the same side conditions as above. Then the λ_m -reduction \rightarrow_m is given as λ_r -reduction, replacing R by F^m and \equiv by \equiv_m in the rules for \rightarrow_r , and modifying the fetch rule as follows:

$$\frac{E \langle F/x \rangle \rightsquigarrow E'}{E \langle F^{m+1}/x \rangle \rightarrow E' \langle F^m/x \rangle}$$

(again, with the same side condition as for the fetch rule). Obviously, deadlocks are still possible when evaluating λ_m -terms. On the other hand, this would be avoided if any resource from a bag were available at will. Then another sub-calculus of Λ_r that naturally arises is the λ -calculus with “infinite resources” $\Lambda_{\infty r}$, given by the grammar:

$$\begin{aligned} E & ::= x \mid \lambda x E \mid (ER) \mid E\langle R/x \rangle \\ R & ::= E^\infty \mid (R \mid R) \end{aligned}$$

Internal choice is obviously definable in this calculus:

$$(E \oplus F) = x\langle E^\infty \mid F^\infty /x \rangle \quad x \notin \text{fv}(E) \cup \text{fv}(F)$$

Indeed, $\Lambda_{\infty r}$ should not be different from Λ_\oplus , the usual λ -calculus enriched with internal choice (see [5]). Therefore, while evaluation in Λ_m is deterministic with potential deadlocks, in $\Lambda_{\infty r}$ it is non deterministic, without deadlocks. The intersection of the two sub-calculi, denoted Λ_∞ , is given by the grammar

$$E ::= x \mid \lambda x E \mid (EE^\infty) \mid E\langle E^\infty /x \rangle$$

This last sub-calculus may be regarded as “the λ -calculus”, with explicit resources. Indeed, there is an obvious translation from Λ to Λ_∞ , namely:

$$\begin{aligned} \mathcal{G}[x] & = x \\ \mathcal{G}[\lambda x M] & = \lambda x \mathcal{G}[M] \\ \mathcal{G}[MN] & = (\mathcal{G}[M] \mathcal{G}[N]^\infty) \end{aligned}$$

In [6] we showed that this translation is fully abstract. To see what this means, we first have to define the observational semantics.

5 The observational semantics

We have seen that the evaluation of a given λ_r -term may diverge, deadlock or terminate properly, and that these possibilities are not mutually exclusive. Then the notion of observability may be richer than in the λ -calculus, and this gives us some freedom in defining the observational semantics of our calculus.

Since we are dealing with “may testing” for the π -calculus, we shall keep this kind of semantics for our λ -calculus with resources. This is much weaker than the bisimulation semantics usually adopted when dealing with a non deterministic calculus (see [24], for instance), though we shall see that, as far as the λ -calculus is concerned, may testing is as fine as bisimulation. As we suggested, computations in Λ_r may be observed in several ways. To formalize this point, let us introduce a domain of *observations* $D = \{\perp, \delta, \gamma\}$, the element of which respectively represent divergence, deadlock and proper termination, that is convergence to a value. In our calculus, a *value* is a closure, that is a term given by the grammar:

$$V ::= \lambda x E \mid V\langle P/x \rangle$$

Let \mathcal{V} denote the set of values. A *deadlock* is a closed term F which is a normal form, but not a value, that is $\{F' \mid F \rightarrow F'\} = \emptyset$ and $F \notin \mathcal{V}$. We denote by \mathcal{W} the set of deadlocks. Then we define the observation function $\text{obs}(E)$, associating a set of observations with any closed term E , as follows:

1. $\perp \in \text{obs}(E)$ if and only if E may diverge, that is there exists an infinite sequence $(E_n)_{n \in \mathbb{N}}$ of terms such that $E_0 = E$ and $E_n \rightarrow_r E_{n+1}$ for any $n \in \mathbb{N}$,
2. $\delta \in \text{obs}(E)$ if and only if E may deadlock, that is $\exists F \in \mathcal{W}. E \xrightarrow{*}_r F$,
3. $\gamma \in \text{obs}(E)$ if and only if E may converge, that is $\exists V \in \mathcal{V}. E \xrightarrow{*}_r V$.

For instance, if $E \in \Lambda_m$ then $\text{obs}(E)$ is a singleton, and if $E \in \Lambda_{\infty r}$ then $\text{obs}(E) \subseteq \{\perp, \gamma\}$. To define the observational semantics, we must have some means to compare the elements of D . Given a preorder on D , we say that E is observationally better than F if, for any λ_r -context C closing both E and F , and for any observation $o \in \text{obs}(C[F])$, there exists an observation $o' \in \text{obs}(C[E])$ which is better than o .

Several observation scenarios, that is preorders on D , are possible – taking Scott’s view that divergence provides no information, that is any observation is better than \perp . In the “standard” scenario, deadlock is not distinguished from divergence. The corresponding observational semantics has the usual definition in this case, based on the observability predicate $E \Downarrow$ meaning that E may have a value. This is the semantics we adopted in [6, 8] for the λ -calculus with multiplicities Λ_m . We showed that, for this calculus, this semantics is the same as the one we would get by adopting a different scenario, where δ is strictly in between \perp and γ , see [9].

In [8] we characterized the standard semantics induced by λ_m -contexts over λ -terms, showing that it is strictly weaker than L evy’s semantics. Since here we are dealing with the relationships between the λ and π calculi, it is appropriate to consider another observation scenario, called the “flat” scenario in [9]. This is given by the *flat ordering* \leq on D , that is:



Definition 5.1 (Observational Semantics) *The observational semantics of the λ_r -calculus is the preorder defined as follows:*

$$E \sqsubseteq_r F \Leftrightarrow \text{for all } \lambda_r\text{-contexts } C, \forall o \in \text{obs}(C[F]) \exists o' \in \text{obs}(C[E]) \quad o \leq o'$$

It is implicitly assumed in this definition that $C[E]$ and $C[F]$ are both closed, since obs is only defined on closed terms. Note that it would be slightly more accurate to deal with an observation function that does not assign any value to divergence, since

$$E \sqsubseteq_r F \Leftrightarrow \text{for all } \lambda_r\text{-contexts } C, \text{obs}'(C[E]) \subseteq \text{obs}'(C[F])$$

where $\text{obs}'(C[E]) = \text{obs}(C[E]) - \{\perp\}$. By restricting the contexts to be λ_m , or $\lambda_{\infty r}$, or λ_{∞} -contexts in the definition, we get corresponding preorders \sqsubseteq_m , $\sqsubseteq_{\infty r}$ and \sqsubseteq_{∞} respectively. Note that in the two latter cases, the observational semantics has the standard definition: E is less than F if, for all contexts C , if $C[E]$ may converge then $C[F]$ may converge. We denote by \simeq_r , \simeq_m , $\simeq_{\infty r}$ and \simeq_{∞} the respective associated equivalences. There is an obvious ordering relation between these semantics: a broader class of contexts determines a stronger semantics. In [6] we showed:

$$\forall M, N \in \Lambda. M \sqsubseteq_{\ell} N \Leftrightarrow \mathcal{G}[M] \sqsubseteq_{\infty} \mathcal{G}[N]$$

This justifies our claim that Λ_{∞} is “the (weak) λ -calculus”. Therefore, we shall omit the translation \mathcal{G} , and regard λ -terms as terms of Λ_r , with implicit infinite multiplicities. This allows us to write $M \sqsubseteq_r N$, $M \sqsubseteq_m N$ and $M \sqsubseteq_{\infty r} N$ for M and N in Λ .

Let us see some examples. One can see that η -expansion on λ -terms is neither increasing nor decreasing in general with respect to the observational semantics. For instance, $x \not\sqsubseteq_r \lambda y(xy)$ and $\lambda y(xy) \not\sqsubseteq_r x$ because if we let $C = \llbracket \langle 0/x \rangle \rrbracket$ then $\text{obs}(C[x]) = \{\delta\}$ while $\text{obs}(C[\lambda x(xy)]) = \{\gamma\}$. Similarly, let $\Xi = (\lambda fx. ff)(\lambda fx. ff)$. Then Ξ is not comparable with the identity \mathbf{I} , since $\text{obs}(\mathbf{I} 0) = \{\delta\}$ while $\text{obs}(\Xi 0) = \{\gamma\}$. These examples show that \sqsubseteq_m is strictly stronger than \sqsubseteq_{ℓ} on λ -terms.

Due to the universal quantification over contexts, the definition of the observational semantics is not very manageable: it is usually quite difficult to prove an inequation $E \sqsubseteq_r F$. In [16], Milner stated and proved a property called the *context lemma*, which was then generalized to the λ -calculus by Lévy [14], establishing that, in order to “test” a (closed) λ -term, it is enough to apply it. The context lemma also holds in the λ -calculus with resources. To prove this fact, we first introduce a restricted kind of contexts, the *applicative contexts*, ranged over by $K, L \dots$. These are given by the grammar:

$$K ::= [] \mid (KP) \mid K\langle P/x \rangle$$

where P is any bag. Note that $E \rightarrow_r F \Rightarrow K[E] \rightarrow_r K[F]$. Now we define the *applicative testing* preorder, which is the observational preorder restricted to applicative contexts, that is:

$$E \sqsubseteq_r^A F \stackrel{\text{def}}{\Leftrightarrow} \forall K. \forall o \in \text{obs}(K[E]) \exists o' \in \text{obs}(K[F]) \quad o \leq o'$$

In [8], we proved the following:

Lemma 5.2 *Let $E, F \in \Lambda_r$, and let $z \notin \text{fv}(E) \cup \text{fv}(F)$. Then*

$$E \sqsubseteq_r^A F \Rightarrow [z/x]E \sqsubseteq_r^A [z/x]F$$

Now we prove the context lemma.

Lemma 5.3 (The Context Lemma)

$$E \sqsubseteq_r F \Leftrightarrow E \sqsubseteq_r^A F$$

PROOF: The direction “ \Rightarrow ” is obvious. To establish “ \Leftarrow ”, we use the notion of a *multiple context*, that is the notion of a context where there may be several kinds of holes, indexed by positive integers, i.e. $[\]_i$. For any such context involving only holes whose indexes are less than k , we define $C[E_1, \dots, E_k]$ in the obvious way, that is by filling the hole $[\]_i$ by the corresponding term E_i . We shall also use the notation $C[\tilde{E}]$ for $C[E_1, \dots, E_k]$. Let \mathcal{S} be the relation consisting of the pairs of closed terms of the form $(C[E_1, \dots, E_k], C[F_1, \dots, F_k])$ where $E_i \sqsubseteq_r^A F_i$ for any i . We show the following:

1. if ESF and $E \xrightarrow{l} E' \in \mathcal{V}$ then $F \xrightarrow{*} F'$ for some $F' \in \mathcal{V}$,
2. if ESF and $E \xrightarrow{l} E' \in \mathcal{W}$ then $F \xrightarrow{*} F'$ for some $F' \in \mathcal{W}$.

where $E \xrightarrow{l} E'$ means that E reduces in l steps to E' . We proceed by induction on (l, h) , w.r.t. the lexicographic ordering, where h is the number of occurrences of holes in C . We may write $C = C_0 C_1 \dots C_m$ where C_0 is either a hole $[\]_i$, or a variable x , or an abstraction context $\lambda x B$, and the C_j 's, for $j > 0$, are bags or substitution contexts. We examine the possible cases (this proof technique is directly adapted from L evy's one [14], with the notable difference that we are dealing with open terms).

(1) if $C_0 = \lambda x B$, we have, by pushing the substitutions on the left while possibly renaming the variables that are bound by these substitutions:

$$C[\tilde{E}] \equiv (\lambda x B[\tilde{G}])B_1[\tilde{G}] \dots B_p[\tilde{G}] \langle D_1[\tilde{G}]/x_1 \rangle \dots \langle D_q[\tilde{G}]/x_q \rangle$$

where $\tilde{G} = G_1, \dots, G_s$ and the G_j 's are obtained from the E_i 's by renaming some free variables by fresh ones (note that in performing this transformation on $C[\tilde{E}]$ we may have replaced some holes $[\]_i$ of C by new holes $[\]_j$, in which a particular renaming of E_i will be put). There are two subcases. If $p = 0$ then $C[\tilde{E}]$ is a value, for any \tilde{E} . Otherwise ($p > 0$), we have $l > 0$ since $C[\tilde{E}]$ is neither a value nor a deadlock. Let

$$C' = B \langle B_1/x \rangle B_2 \dots B_p \langle D_1/x_1 \rangle \dots \langle D_q/x_q \rangle$$

Then

$$C[\tilde{E}] \rightarrow C'[\tilde{G}] \xrightarrow{l-1} E'$$

and

$$C[\tilde{F}] \rightarrow C'[\tilde{H}]$$

where H_1, \dots, H_s are obtained from the F_i 's by the same renamings as the one we used to get the G_j 's from the E_i 's. Then one uses the previous lemma, and the induction hypothesis to conclude.

(2) if $C_0 = x$, we have, as in the previous case

$$C[\tilde{E}] \equiv z B_1[\tilde{G}] \dots B_p[\tilde{G}] \langle D_1[\tilde{G}]/x_1 \rangle \dots \langle D_q[\tilde{G}]/x_q \rangle$$

Let i be the first index such that $x_i = z$ (such an i exists since $C[\tilde{E}]$ is closed). There are again two subcases. If $D_i[\tilde{G}] \equiv 0$ then $C[\tilde{E}]$ is a deadlock. Since 0 is not a λ_r -term, we have $D_i[\tilde{H}] \equiv 0$ for any \tilde{H} , therefore $C[\tilde{F}]$ is a deadlock too. Otherwise, $D_i[\tilde{G}] \equiv (B[\tilde{G}] \mid D'_i[\tilde{G}])$ where $B[\tilde{G}] \in \Lambda_r$, so that if we let

$$C' = B B_1 \cdots B_p \langle D_1/x_1 \rangle \cdots \langle D'_i/x_i \rangle \cdots \langle D_q/x_q \rangle$$

we have

$$C[\tilde{E}] \rightarrow C'[\tilde{G}] \xrightarrow{l-1} E'$$

Since

$$C[\tilde{F}] \rightarrow C'[\tilde{H}]$$

we may conclude as in the previous case.

(3) if $C_0 = []_i$, let $C' = E_i C_1 \cdots C_m$. This context has $h - 1$ holes, and obviously $C'[\tilde{E}] = C[\tilde{E}]$, therefore by induction hypothesis if $C[\tilde{E}] \xrightarrow{l} E'$ with $E' \in \mathcal{V}$ or $E' \in \mathcal{W}$, then there exists F'' such that $F'' \in \mathcal{V}$ or $F'' \in \mathcal{W}$ respectively, and $C'[\tilde{F}] \xrightarrow{*} F''$. Since $C'[\tilde{F}] = E_i C_1[\tilde{F}] \cdots C_m[\tilde{F}]$ and $E_i \sqsubseteq_r^A F_i$, we conclude that $C[\tilde{F}] \xrightarrow{*} F'$ for some F' such that $F' \in \mathcal{V}$ or $F' \in \mathcal{W}$ respectively. ■

Remark 5.4 Using the Context Lemma it is easy to verify that $\Omega \sqsubseteq_r E$ for any E . Then for instance

$$m \leq n \Rightarrow \lambda x_1 \cdots x_m. \Omega \sqsubseteq_r \lambda x_1 \cdots x_n. \Omega$$

Another consequence of the Context Lemma we shall use is the following:

Lemma 5.5

- (i) For any $E, F \in \Lambda_r$, $E \rightarrow_r F \Rightarrow F \sqsubseteq_r E$. Moreover, $(\lambda x E)P \simeq_r E\langle P/x \rangle$.
- (ii) For any $E, F \in \Lambda_m$, $E \rightarrow_m F \Rightarrow F \simeq_r E$.

PROOF: the first point should be obvious, since $E \rightarrow_r F \Rightarrow K[E] \rightarrow_r K[F]$ for any applicative context K , therefore $o \in \text{obs}(K[F]) \Rightarrow o \in \text{obs}(K[E])$. For the second point, note that if $o \in \text{obs}(K[(\lambda x E)P])$ then also $o \in \text{obs}(K[E\langle P/x \rangle])$ since any non-empty reduction sequence originating in $K[(\lambda x E)P]$ must start with $K[(\lambda x E)P] \rightarrow_r K[E\langle P/x \rangle]$. The last point is shown in a similar way, that is, using the fact that λ_m -reduction is deterministic. ■

Note that the first implication cannot be reversed in general, because λ_r -reduction is non deterministic. It is an easy exercise to show, using the Context Lemma, that internal choice is a *join*:

$$(E_0 \oplus E_1) \sqsubseteq_r E \Leftrightarrow E_0 \sqsubseteq_r E \ \& \ E_1 \sqsubseteq_r E$$

Corollary 5.6

- (i) For any $M, N \in \Lambda$ $M =_\beta N \Rightarrow M \simeq_r N$.
- (ii) Moreover, if $A \in \mathcal{A}(M)$ then $A \sqsubseteq_r M$.

PROOF: The first point is an immediate consequence of the previous lemma. Regarding the second, if $A \in \mathcal{A}(M)$ then by definition there is $N =_\beta M$ such that $A = \varpi(N)$. We have $N \simeq_r M$, and $A \sqsubseteq_r N$ since \sqsubseteq_r is a precongruence such that $\Omega \sqsubseteq_r X$ for any X (see the previous remark). ■

6 The discriminating power of multiplicities

In this section we characterize the semantics induced *over λ -terms* by the λ -calculi with resources and multiplicities. We show that, for $M, N \in \Lambda$, $M \sqsubseteq_r N$ and $M \sqsubseteq_m N$ both coincide with L evy's ordering $M \sqsubseteq_{\mathcal{L}} N$.

As a first step towards these results, we establish a property that we call the *approximation lemma* (cf. [14]). It states that, to observe $C[M]$ in some way (where M is a λ -term and C a λ_r -context), only a finite amount of information about M needs to be known. Intuitively, this should be clear, because M can only participate by a finite number of reduction steps in a computation of $C[M]$. Moreover, it is only when M shows up in the head position, as a function applied to a series of arguments, that it has to exhibit some specific finite intensional content, like beginning with a series of abstractions. Then any term having at least the same intensional content is as good as M , as far as the observability within the context C is concerned. The appropriate formalization of "finite intensional content" is given by approximants.

Lemma 6.1 (The Approximation Lemma) *For any λ_r -context C and for every $M \in \Lambda$ with $C[M]$ closed:*

$$o \in \text{obs}(C[M]) \Rightarrow \exists A \in \mathcal{A}(M) \exists o' \geq o. \quad o' \in \text{obs}(C[A])$$

PROOF: We use multiple contexts, as in the context lemma (again, the explicit substitution construct is very convenient for this proof). Firstly, we note that if $o = \perp$ then one can take Ω as the appropriate approximant for any M_i (recall that $\Omega \in \mathcal{A}(M)$ for any M). Then we show the following:

1. if $C[M_1, \dots, M_k] \xrightarrow{l} V \in \mathcal{V}$ then there exists A_1, \dots, A_k such that $A_i \in \mathcal{A}(M_i)$ for any i and $C[A_1, \dots, A_k] \xrightarrow{*} V'$ for some $V' \in \mathcal{V}$,
2. if $C[M_1, \dots, M_k] \xrightarrow{l} W \in \mathcal{V}$ then there exists A_1, \dots, A_k such that $A_i \in \mathcal{A}(M_i)$ for any i and $C[A_1, \dots, A_k] \xrightarrow{*} W'$ for some $W' \in \mathcal{V}$.

The proof is entirely similar to the one of the Context Lemma 5.3. The details are left to the reader. You should use the following facts:

1. $A \in \mathcal{A}(M) \Rightarrow [z/x]A \in \mathcal{A}([z/x]M)$. This is needed because some of the M_i 's may have to be duplicated and renamed.

2. $\mathcal{A}(M)$ is directed (with respect to the prefix ordering \preceq , see the Lemma 3.2). This is needed because some of the M_i 's may occur in different positions in $C[\widehat{M}]$, where different approximants may be used.
3. $A \preceq B \Rightarrow A \sqsubseteq_r B$ (see the Corollary 5.6).

A complete proof of the same result, though for a different observation scenario, may be found in [8]. ■

A consequence of this lemma, together with the Corollary 5.6(ii) is that $M \sqsubseteq_{\mathcal{L}} N$ implies $M \sqsubseteq_r N$ (see the Theorem 6.4 below). To prove the converse, the key result is a *separation lemma*, showing that if M and N differ in Lévy's interpretation, then there is a context with multiplicities separating these two terms. To prove this result, it is convenient to use an alternative characterization of Lévy's preorder, as the "limit" of a decreasing sequence of preorders.

Proposition 6.2

$$M \sqsubseteq_{\mathcal{L}} N \Leftrightarrow \forall k \in \mathbb{N}. M \leq_k N$$

where \leq_k , the intensional preorder at order k , is given by

1. $M \leq_0 N$ for any M and N ;
2. $M \leq_{k+1} N$ if and only if
 - (a) $M \in \text{PO}_n$ and $N =_{\beta} \lambda x_1 \cdots x_m. N'$ with $m \geq n$, or
 - (b) $M =_{\beta} \lambda x_1 \dots x_n. x M_1 \cdots M_s$ and $N =_{\beta} \lambda x_1 \dots x_n. x N_1 \cdots N_s$ with $M_i \leq_k N_i$ for $1 \leq i \leq s$.

The separation lemma will establish that if M and N intensionally differ at some finite order, that is $M \not\leq_k N$ for some k , then one can test the difference in the λ_m -calculus. That is, there is a λ_m -context C *separating* these two terms, in the sense that $\text{obs}(C[M]) \not\leq \text{obs}(C[N])$ (this makes sense since both $\text{obs}(C[M])$ and $\text{obs}(C[N])$ are singletons in this case). The proof, by induction on k , uses a refinement of the classical "Böhm-out technique" (see [4]). As such, it uses the *tupling* combinators

$$\mathbf{P}_n = \lambda x_1 \cdots x_{n+1}. x_{n+1} x_1 \cdots x_n$$

and the *projections*

$$\mathbf{U}_i^n = \lambda x_1 \cdots x_n. x_i$$

Lemma 6.3 (The Separation Lemma) *Let $M, N \in \Lambda$ and $\text{fv}(M) \cup \text{fv}(N) \subseteq \{x_1, \dots, x_n\}$. If $M \not\leq_k N$ then there exist p_1, \dots, p_n and m_1, \dots, m_n in \mathbb{N} such that for any q_1, \dots, q_n with $q_i \geq p_i$ there exist closed bags P_1, \dots, P_r such that*

$$\text{obs}(M \langle \mathbf{P}_{q_1}^{m_1} / x_1 \rangle \cdots \langle \mathbf{P}_{q_n}^{m_n} / x_n \rangle P_1 \cdots P_r) \not\leq \text{obs}(N \langle \mathbf{P}_{q_1}^{m_1} / x_1 \rangle \cdots \langle \mathbf{P}_{q_n}^{m_n} / x_n \rangle P_1 \cdots P_r).$$

PROOF: By induction on the least k such that $M \not\leq_k N$. In the proof we shall use $\Sigma, \Xi \dots$ to denote sequences of substitutions $\langle P_1/x_1 \rangle \dots \langle P_n/x_n \rangle$. More precisely, when we write $E\Sigma$ where $\Sigma = \langle P_1/x_1 \rangle \dots \langle P_n/x_n \rangle$ this must be read $(\dots(E\langle P_1/x_1 \rangle)) \dots \langle P_n/x_n \rangle$.

(1) $k = 1$. The case $M =_{\beta} \lambda z_1 \dots z_m. \Omega$ and $N =_{\beta} \lambda y_1 \dots y_h. N' \Rightarrow h < m$ is easy: we let $m_i = 0$ and $p_i = 0$ for any i , $r = h$ and $P_j = \Omega^{\infty}$.

If $M =_{\beta} \lambda z_1 \dots z_m. x M_1 \dots M_s$ and $N =_{\beta} \lambda y_1 \dots y_h. \Omega$, there are two subcases, according as $x = x_i$ or $x = z_i$, for some i . In the first case, we let $m_j = p_j = 0$ for $j \neq i$, $m_i = 1$, $p_i = s + r - m$ where $r = \max\{h, m\}$ and $P_j = \Omega^{\infty}$ for $1 \leq j \leq r$. Otherwise, if $x = z_i$ then we let $m_j = 0$ for every j , $P_j = \Omega^{\infty}$ for $j \neq i$ and $P_i = (\lambda y_1 \dots y_u. y_u)^{\infty}$, where $u = s + r - m + 1$. We let the reader check that these are appropriate choices.

If $M =_{\beta} \lambda z_1 \dots z_m. x M_1 \dots M_s$ and $N =_{\beta} \lambda z_1 \dots z_h. z N_1 \dots N_t$, then there are several subcases.

(1.1) $m \neq h$. We only examine the case $m < h$, the other one being similar. If $x = x_i$ the lemma follows by taking $m_j = 0$ for any j , and $r = m$ (P_j may be any bag). If $x = z_i$ then we let $m_j = 0$ for any j , $P_i = \Omega^0$ while for $j \neq i$, P_j may be any bag.

(1.2) $m = h$. Then we have $x \neq z$ or $s \neq t$, and $x = x_i$ or $x = z_i$, for some i . We just examine one case, where $t < s$ and $x = x_i = z$, the other ones being similar. We let $m_j = 0$ and $p_j = 0$ for $j \neq i$ and $m_i = 1$, $p_i = s + h - m$. Now if q_1, \dots, q_n are such that $q_j \geq p_j$ for any j , we let $r = h + q_i - t + 1$, $P_j = \Omega^{\infty}$ for $j \neq h + q_i - p_i + 1$ (note that $h + q_i - p_i + 1 < r$ since $t < p_i$) and $P_{h+q_i-p_i+1} = (\lambda y_1 \dots y_l. y_l)^{\infty}$ where $l = q_i + p_i - t + 1$. Let $\Sigma = \langle \mathbf{P}_{q_1}^0/x_1 \rangle \dots \langle \mathbf{P}_{q_n}^0/x_n \rangle$ and $\Gamma = \langle P_1/z_1 \rangle \dots \langle P_m/z_m \rangle$. We have – counting only the number of arguments, not of substitutions, in the underbraced parts, and omitting the infinite multiplicities:

$$\begin{aligned}
& (M \langle \mathbf{P}_{q_1}^{m_1}/x_1 \rangle \dots \langle \mathbf{P}_{q_n}^{m_n}/x_n \rangle) P_1 \dots P_r \\
& \xrightarrow{*}_m (\lambda z_1 \dots z_m. (\mathbf{P}_{q_i} M_1 \dots M_s) \Sigma) P_1 \dots P_r \\
& \xrightarrow{*}_m ((\mathbf{P}_{q_i} M_1 \dots M_s) \underbrace{\Sigma \Gamma}_{p_i} P_{m+1} \dots P_{h+1} \dots \underbrace{P_{h+q_i-p_i+1}}_{q_i-p_i+1} \dots \underbrace{P_r}_{p_i-t}) \\
& \xrightarrow{*}_m ((P_{h+q_i-p_i+1} \underbrace{M_1 \dots M_s \Omega \dots \Omega}_{q_i+p_i-t} \Sigma \Gamma) \Omega \dots \Omega) \simeq_m \mathbf{I}
\end{aligned}$$

while

$$\begin{aligned}
 & (N\langle \mathbf{P}_{q_1}^{m_1}/x_1 \rangle \cdots \langle \mathbf{P}_{q_n}^{m_n}/x_n \rangle) P_1 \cdots P_r \\
 & \xrightarrow{*}_m (\lambda z_1 \dots z_h \cdot (\mathbf{P}_{q_i} N_1 \cdots N_t) \Sigma) P_1 \cdots P_r \\
 & \xrightarrow{*}_m ((\mathbf{P}_{q_i} \underbrace{N_1 \cdots N_t}_{q_i - p_i + t + 1}) \Sigma \Gamma') \underbrace{P_{h+1} \cdots P_{h+q_i - p_i + 1}}_{p_i - t} \cdots P_r \\
 & \xrightarrow{*}_m ((\Omega N_1 \cdots N_t \Omega \cdots \Omega) \Sigma \Gamma') \Omega \cdots P_{h+q_i - p_i + 1} \cdots \Omega \simeq_m \Omega
 \end{aligned}$$

where $\Gamma' = \langle P_1/z_1 \rangle \cdots \langle P_h/z_h \rangle$.

(2) $k > 1$. The only possibility here is $M =_{\beta} \lambda z_1 \dots z_m \cdot x M_1 \cdots M_s$ and $N =_{\beta} \lambda z_1 \dots z_m \cdot x N_1 \cdots N_s$ with $M_l \not\leq_{k-1} N_l$, for some l . There are two sub-cases, according as $x = x_i$ or $x = z_i$ for some i . We only examine the second one. We have $\text{fv}(M_l) \cup \text{fv}(N_l) \subseteq \{x_1, \dots, x_n\} \cup \{z_1, \dots, z_m\}$. By induction hypothesis, there exist $\pi_1, \dots, \pi_n, \pi'_1, \dots, \pi'_m$ and $\mu_1, \dots, \mu_n, \mu'_1, \dots, \mu'_m$ such that for any $\kappa_1, \dots, \kappa_n, \kappa'_1, \dots, \kappa'_m$ with $\kappa_j \geq \pi_j$ and $\kappa'_j \geq \pi'_j$ there are Q_1, \dots, Q_ρ with

$$\text{obs}(M_l \Sigma \Gamma Q_1 \cdots Q_\rho) \not\leq \text{obs}(N_l \Sigma \Gamma Q_1 \cdots Q_\rho).$$

where $\Sigma = \langle \mathbf{P}_{\kappa_1}^{\mu_1}/x_1 \rangle \cdots \langle \mathbf{P}_{\kappa_n}^{\mu_n}/x_n \rangle$, and $\Gamma = \langle \mathbf{P}_{\kappa'_1}^{\mu'_1}/z_1 \rangle \cdots \langle \mathbf{P}_{\kappa'_m}^{\mu'_m}/z_m \rangle$. Then the proof consists in finding a context C such that $C[M]$ is essentially $M_l \Sigma \Gamma Q_1 \cdots Q_\rho$, while $C[N]$ gives $N_l \Sigma \Gamma Q_1 \cdots Q_\rho$.

Let $m_j = \mu_j$ for $j \neq i$ and $m_i = \mu_i + 1$, $p_j = \pi_j$ for $j \neq i$ and $p_i = \max(\pi_i, s + t)$. Given q_1, \dots, q_n such that $q_j \geq p_j$ for any j , let Q_1, \dots, Q_ρ be a sequence satisfying the property above for $\kappa'_j = \pi'_j$ and $\kappa''_j = \pi''_j$. Take $r = \rho + h$ where $h = q_i + 1 + m - s$, and P_1, \dots, P_r be the sequence defined as follows:

$$P_j = \begin{cases} \mathbf{P}_{\pi'_j}^{\mu'_j} & \text{if } 1 \leq j \leq m \\ \mathbf{P}_{\pi''_j - m}^{\mu''_j - m} & \text{if } m < j \leq t \\ (\lambda u_1 \dots u_{q_i} \cdot u_i)^\infty & \text{if } j = h \\ Q_{j-h} & \text{if } h < j \leq r \\ \Omega^\infty & \text{otherwise} \end{cases}$$

This sequence is thus – still omitting the infinite multiplicities:

$$P_1 \cdots P_r = \mathbf{P}_{\pi'_1}^{\mu'_1} \cdots \mathbf{P}_{\pi'_m}^{\mu'_m} \mathbf{P}_{\pi''_1}^{\mu''_1} \cdots \mathbf{P}_{\pi''_t}^{\mu''_t} \underbrace{\Omega \cdots \Omega}_{q_i - (s+t)} P_h Q_1 \cdots Q_\rho$$

We leave to the reader to check that

$$(M\langle \mathbf{P}_{q_1}^{m_1}/x_1 \rangle \cdots \langle \mathbf{P}_{q_n}^{m_n}/x_n \rangle) P_1 \cdots P_r \xrightarrow{*}_m (M_l \Sigma \Xi \Gamma) Q_1 \cdots Q_\rho$$

and

$$(N\langle \mathbf{P}_{q_1}^{m_1} / x_1 \rangle \cdots \langle \mathbf{P}_{q_n}^{m_n} / x_n \rangle) P_1 \cdots P_r \xrightarrow{*}_m (N_l \Sigma \Xi \Gamma) Q_1 \cdots Q_\rho .$$

This completes the proof. ■

In [8] we proved, in a more detailed way, a similar result for the ‘‘standard’’ observation scenario. The situation here is simpler, because the semantics \sqsubseteq_r is much more discriminating than the standard one, therefore it is easier to separate terms by distinguishing deadlock from divergence. Now we can prove the main result of this section:

Theorem 6.4 *For any $M, N \in \Lambda$*

$$M \sqsubseteq_{\mathcal{L}} N \Leftrightarrow M \sqsubseteq_r N \Leftrightarrow M \sqsubseteq_m N$$

PROOF: Assume that $M \sqsubseteq_{\mathcal{L}} N$, and let C be a λ_r -context such that $o \in \text{obs}(C[M])$. By the Approximation Lemma 6.1, there exist $A \in \mathcal{A}(M)$ and $o' \geq o$ such that $o' \in \text{obs}(C[A])$. We have $A \in \mathcal{A}(N)$, therefore by the Corollary 5.6 there exists $o'' \geq o'$ such that $o'' \in \text{obs}(C[N])$. This shows $M \sqsubseteq_{\mathcal{L}} N \Rightarrow M \sqsubseteq_r N$. The implication $M \sqsubseteq_r N \Rightarrow M \sqsubseteq_m N$ is trivial. Finally the implication $M \sqsubseteq_m N \Rightarrow M \sqsubseteq_{\mathcal{L}} N$ is the Separation Lemma. ■

7 Relating the calculi

In this section we show that, as far as λ -terms are concerned, the π , λ_r and λ_m -semantics all coincide with L evy’s interpretation. This is our main result. In order to prove it, we relate the π , λ_r and λ_m -calculi. Let us first see how the λ_r -calculus may be encoded into the π -calculus.

The idea of the encoding is quite simple: we have seen that $\mathcal{E}[(\lambda x M)N]u$ reduces to $(\nu x)(\mathcal{E}[M]u \mid \mathcal{E}[x := N])$, a term that might be taken as representing $[N/x]M$. This gives us the way we shall encode explicit substitutions $E\langle P/x \rangle$. It just remains to encode the bags, or more precisely the substitution items $\langle P/x \rangle$. The syntax we chose for bags should indicate how to proceed. Here is the encoding:

$$\begin{aligned} \mathcal{B}[x]u &= \bar{x}u \\ \mathcal{B}[\lambda x E]u &= u(x, v)\mathcal{B}[E]v \\ \mathcal{B}[ER]u &= (\nu v)(\mathcal{B}[E]v \mid (\nu x)(\bar{v}xu \mid \mathcal{B}[x := R])) \\ \mathcal{B}[E\langle R/x \rangle]u &= (\nu x)(\mathcal{B}[E]u \mid \mathcal{B}[x := R]) \\ \mathcal{B}[x := 0] &= 0 \\ \mathcal{B}[x := E] &= x(w)\mathcal{B}[E]w \\ \mathcal{B}[x := (P \mid Q)] &= (\mathcal{B}[x := P] \mid \mathcal{B}[x := Q]) \\ \mathcal{B}[x := E^\infty] &= !\mathcal{B}[x := E] \end{aligned}$$

Clearly, Milner's encoding from Λ to Π factorizes into the translation from Λ to Λ_∞ composed with the previous encoding:

$$\forall M \in \Lambda. \mathcal{E}[[M]]u = \mathcal{B}[\mathcal{G}[[M]]]u$$

Note also that the encoding of internal choice is exactly the one we would expect:

$$\mathcal{B}[[E \oplus F]]u = (\nu x)(\bar{x}u \mid x(w)\mathcal{B}[[E]]w \mid x(w)\mathcal{B}[[F]]w)$$

Regarding the correspondence between the notions of reduction, the situation is now better than in the case of λ versus π : we can show that the reduction steps are exactly mimicked in both directions. Actually, the λ_r -calculus was designed exactly for this purpose. In particular, we could show the following:

Lemma 7.1 *For any closed λ_r -term E*

$$\mathcal{B}[[E]]u \Downarrow_\pi \Leftrightarrow \gamma \in \text{obs}(E)$$

That is, one can test in the π -calculus whether a closed λ_r -term has a value or not. This shows that this encoding is adequate with respect to the “standard” observational semantics, where deadlocks are not observed.

However, this is not enough for our purpose, since we want to be able to detect also the potential deadlocks using π -calculus means. To this end, we could modify the observational semantics of the π -calculus, allowing deadlocks to be observable in this calculus (in the π -calculus, a deadlock is a term that does not perform any transition, labelled or unlabelled). This is not the way we shall follow here, since we are interested in the observational semantics as it was given by Milner in [18].

We shall instead establish that it is possible to detect deadlocks arising in the λ -calculus with multiplicities Λ_m , using the π -calculus as it is. Intuitively, the reason is this: in the λ_m -calculus, since the resources are always copies of a given term, one may assume that they are consumed sequentially, that is, without using the commutativity of bag composition. Then one may modify the previous encoding accordingly: a substitution item $\langle E^m/x \rangle$ (where $m \in \mathbb{N}$) is not encoded as a parallel composition of m identical items $\langle E^1/x \rangle$, but as a “stack” of resources, and a specific signal is emitted when no resource is available anymore, that is when there is a potential deadlock. Let us assume that δ and γ are names, belonging to \mathcal{N} . We define the encoding \mathcal{B}_δ from Λ_m into Π , as follows:

$\mathcal{B}_\delta[x]u$	$= \bar{x}u$	
$\mathcal{B}_\delta[\lambda x E]u$	$= u(x, v)\mathcal{B}_\delta[E]v$	$v \neq \delta$
$\mathcal{B}_\delta[EF^m]u$	$= (\nu v)(\mathcal{B}_\delta[E]v \mid (\nu x)(\bar{v}xu \mid \mathcal{B}_\delta[x := F^m]))$	$v \neq \delta$
$\mathcal{B}_\delta[E\langle F^m/x \rangle]u$	$= (\nu x)(\mathcal{B}_\delta[E]u \mid \mathcal{B}_\delta[x := F^m])$	
$\mathcal{B}_\delta[x := E^0]$	$= x(w)\bar{\delta}$	$w \neq \delta$
$\mathcal{B}_\delta[x := E^{m+1}]$	$= x(w)(\mathcal{B}_\delta[E]w \mid \mathcal{B}_\delta[x := E^m])$	$w \neq \delta$
$\mathcal{B}_\delta[x := E^\infty]$	$= !x(w)\mathcal{B}_\delta[E]w$	$w \neq \delta$

where $m \in \mathbb{N}$. Obviously, we have, for $M \in \Lambda$

$$\mathcal{E}[M]u = \mathcal{B}_\delta[\mathcal{G}[M]]u$$

The operational correspondence established by this encoding is shown by the following lemma.

Lemma 7.2 *Let E be a closed λ_m -term, and $u \neq \delta$. Then*

- (i) *if $E \equiv_m F$ then $\mathcal{B}_\delta[E]u \equiv_\pi \mathcal{B}_\delta[F]u$,*
- (ii) *if $E \rightarrow_m E'$ then $\mathcal{B}_\delta[E]u \rightarrow_\pi P$ for some P such that $P \equiv_\pi \mathcal{B}_\delta[E']u$,*
- (iii) *if $\mathcal{B}_\delta[E]u \rightarrow_\pi P$ then either E is a deadlock and $P \downarrow \delta$, or $E \rightarrow_m E'$ for some E' such that $P \equiv_\pi \mathcal{B}_\delta[E']u$.*

PROOF: The first point is easily proved, by induction on the definition of $E \equiv_m F$. For the second point, we use the Lemma 4.1. We have

$$E \equiv_m H F_1^{m_1} \dots F_n^{m_n} \langle G_1^{r_1}/x_1 \rangle \dots \langle G_k^{r_k}/x_k \rangle$$

where $H = \lambda x F$ or $H = x$, the variables x_i 's are pairwise distinct, and $i < j \Rightarrow x_i \notin \text{fv}(Q_j)$. Then

$$\begin{aligned} \mathcal{B}_\delta[E]u \equiv_\pi & (\nu u_1 \dots u_n)(\nu z_1 \dots z_n)(\nu x_1 \dots x_k) (\mathcal{B}_\delta[H]u_1 \mid \\ & \bar{u}_1 z_1 u_2 \mid \mathcal{B}_\delta[z_1 := F_1^{m_1}] \mid \\ & \vdots \\ & \bar{u}_n z_n u \mid \mathcal{B}_\delta[z_n := F_n^{m_n}] \mid \\ & \mathcal{B}_\delta[x_1 := G_1^{r_1}] \mid \dots \mid \mathcal{B}_\delta[x_k := G_k^{r_k}]) \end{aligned}$$

If $H = \lambda x F$ we have $\mathcal{B}_\delta[H]u_1 = u_1(x, v)\mathcal{B}_\delta[F]v$, therefore, in this case where

$$E' = F \langle F_1^{m_1}/x \rangle F_2^{m_2} \dots F_n^{m_n} \langle G_1^{r_1}/x_1 \rangle \dots \langle G_k^{r_k}/x_k \rangle$$

we have

$$\begin{aligned} \mathcal{B}_\delta\llbracket E \rrbracket u \rightarrow_\pi (\nu u_1 \dots u_n)(\nu z_1 \dots z_n)(\nu x_1 \dots x_k) & (\mathcal{B}_\delta\llbracket [z_1/x]F \rrbracket u_2 \mid \mathcal{B}_\delta\llbracket z_1 := F_1^{m_1} \rrbracket \\ & \bar{u}_2 z_2 u_3 \mid \mathcal{B}_\delta\llbracket z_2 := F_2^{m_2} \rrbracket \mid \\ & \vdots \\ & \bar{u}_n z_n u \mid \mathcal{B}_\delta\llbracket z_n := F_n^{m_n} \rrbracket \mid \\ & \mathcal{B}_\delta\llbracket x_1 := G_1^{r_1} \rrbracket \mid \dots \mid \mathcal{B}_\delta\llbracket x_k := G_k^{r_k} \rrbracket) \end{aligned}$$

that is $\mathcal{B}_\delta\llbracket E \rrbracket u \rightarrow_\pi P$ with $P \equiv_\pi \mathcal{B}_\delta\llbracket E' \rrbracket u$.

If $H = x_i$, then $r_i > 0$ and $E' = G_i F_1^{m_1} \dots F_n^{m_n} \langle G_1^{r_1}/x_1 \rangle \dots \langle G_i^{r_i-1}/x_i \rangle \dots \langle G_k^{r_k}/x_k \rangle$, we have, since $\mathcal{B}_\delta\llbracket x_i \rrbracket u_1 = \bar{x}_i u_1$ and $\mathcal{B}_\delta\llbracket x_i := G_i^{r_i} \rrbracket = x_i(w)(\mathcal{B}_\delta\llbracket G_i \rrbracket w \mid \mathcal{B}_\delta\llbracket x_i := G_i^{r_i-1} \rrbracket)$

$$\begin{aligned} \mathcal{B}_\delta\llbracket E \rrbracket u \rightarrow_\pi (\nu u_1 \dots u_n)(\nu z_1 \dots z_n)(\nu x_1 \dots x_k) & (\mathcal{B}_\delta\llbracket G_i \rrbracket u_1 \mid \\ & \bar{u}_1 z_1 u_2 \mid \mathcal{B}_\delta\llbracket z_1 := F_1^{m_1} \rrbracket \mid \\ & \vdots \\ & \bar{u}_n z_n u \mid \mathcal{B}_\delta\llbracket z_n := F_n^{m_n} \rrbracket \mid \\ & \mathcal{B}_\delta\llbracket x_1 := G_1^{r_1} \rrbracket \mid \dots \mid \\ & \mathcal{B}_\delta\llbracket x_i := G_i^{r_i-1} \rrbracket \mid \dots \mid \mathcal{B}_\delta\llbracket x_k := G_k^{r_k} \rrbracket) \end{aligned}$$

that is, again, $\mathcal{B}_\delta\llbracket E \rrbracket u \rightarrow_\pi P$ with $P \equiv_\pi \mathcal{B}_\delta\llbracket E' \rrbracket u$.

Regarding the last point, we may assume, as in for the previous point, that E has the following shape:

$$E \equiv_m H F_1^{m_1} \dots F_n^{m_n} \langle G_1^{r_1}/x_1 \rangle \dots \langle G_k^{r_k}/x_k \rangle$$

where $H = \lambda x F$ or $H = x$, the variables x_i 's are pairwise distinct, and $i < j \Rightarrow x_i \notin \text{fv}(Q_j)$. If $\mathcal{B}_\delta\llbracket E \rrbracket u \rightarrow_\pi P$ then either H is an abstraction, or a variable x_i with $r_i > 0$, in which cases it is easy to see that there exists E' such that $E \rightarrow_m E'$ and $P \equiv_\pi \mathcal{B}_\delta\llbracket E' \rrbracket u$, or $H = x_i$ with $r_i = 0$, that is, E is a deadlock, and

$$\begin{aligned} P = (\nu u_1 \dots u_n)(\nu z_1 \dots z_n)(\nu x_1 \dots x_k) & (\bar{\delta} \mid \\ & \bar{u}_1 z_1 u_2 \mid \mathcal{B}_\delta\llbracket z_1 := F_1^{m_1} \rrbracket \mid \\ & \vdots \\ & \bar{u}_n z_n u \mid \mathcal{B}_\delta\llbracket z_n := F_n^{m_n} \rrbracket \mid \\ & \mathcal{B}_\delta\llbracket x_1 := G_1^{r_1} \rrbracket \mid \dots \mid \mathcal{B}_\delta\llbracket x_k := G_{i-1}^{r_{i-1}} \rrbracket \mid \\ & \mathcal{B}_\delta\llbracket x_1 := G_{i+1}^{r_{i+1}} \rrbracket \mid \dots \mid \mathcal{B}_\delta\llbracket x_k := G_k^{r_k} \rrbracket) \end{aligned}$$

and clearly $P \downarrow \delta$. ■

One can now see that if $\mathcal{B}_\delta\llbracket E \rrbracket u \xrightarrow{*}_\pi P \downarrow$ then either E has a value and $P \downarrow u$, or E reduces into a deadlock and $P \downarrow \delta$. Therefore we have:

Corollary 7.3 *For any closed λ_m -term E*

$$(\nu\delta)\mathcal{B}_\delta[E]\gamma\Downarrow_\pi \Leftrightarrow \gamma \in \text{obs}(E) \quad \text{and} \quad (\nu\gamma)\mathcal{B}_\delta[E]\gamma\Downarrow_\pi \Leftrightarrow \delta \in \text{obs}(E)$$

For $E, F \in \Lambda_m$, let us write $E \sqsubseteq_\pi F$ for $\mathcal{B}_\delta[E] \sqsubseteq_\pi \mathcal{B}_\delta[F]$. Note that for λ -terms $M, N \in \Lambda$, this coincides, up to the identification of M with $\mathcal{G}[M]$, with our previous definition of $M \sqsubseteq_\pi N$, that is $\mathcal{E}[M]u \sqsubseteq_\pi \mathcal{E}[N]u$.

Proposition 7.4 *For any $E, F \in \Lambda_m$*

$$E \sqsubseteq_\pi F \Rightarrow E \sqsubseteq_m F$$

PROOF: Let C be a λ_m -context, and $o \in \text{obs}(C[E])$. We have to show that there exists $o' \in \text{obs}(C[F])$ such that $o \leq o'$ and $o' \in \text{obs}(C[F])$. This is obvious if $o = \perp$. If $o = \gamma$, then by the Corollary 7.3 we have $(\nu\delta)\mathcal{B}_\delta[C[E]]\gamma\Downarrow_\pi$. Since the encoding \mathcal{B}_δ is defined in a compositional way, there is a multiple π -context D and names u_1, \dots, u_k such that

$$\mathcal{B}_\delta[C[G]]\gamma = D[\mathcal{B}_\delta[G]u_1, \dots, \mathcal{B}_\delta[G]u_k]$$

for any $G \in \Lambda_m$, therefore $(\nu\delta)\mathcal{B}_\delta[C[F]]\gamma\Downarrow_\pi$ since $E \sqsubseteq_\pi F$, hence $\gamma \in \text{obs}(C[F])$ by the Corollary 7.3. The case $o = \delta$ is similar. ■

We can now prove the announced result:

Theorem 7.5 *For any $M, N \in \Lambda$*

$$M \sqsubseteq_{\mathcal{L}} N \Leftrightarrow M \sqsubseteq_r N \Leftrightarrow M \sqsubseteq_m N \Leftrightarrow M \sqsubseteq_\pi N$$

PROOF: The first two equivalences are given by the Theorem 6.4, $M \sqsubseteq_\pi N \Rightarrow M \sqsubseteq_m N$ is given by the previous proposition, while $M \sqsubseteq_{\mathcal{L}} N \Rightarrow M \sqsubseteq_\pi N$ is given by the Theorem 3.3. ■

8 Conclusion

To conclude this paper, let us briefly comment on our result, with respect to related work. Firstly, we note that the π -may testing equivalence $M \simeq_\pi N$ on λ -terms coincides with the π -bisimulation $M \approx_\pi N$ considered by Sangiorgi (see [24, 25]), since he showed that the latter coincides with equality $M =_{\mathcal{L}} N$ in L evy's interpretation. It is worth recalling here that the π -bisimulation $M \approx_\pi N$ also coincides with the bisimulation $M \approx_\oplus N$ induced by the λ -calculus enriched with internal choice ($M \oplus N$), as shown by Sangiorgi in [24].

The reasons why we preferred to deal with may testing – besides the fact that this is the semantics considered by Milner in [18] – are the following. Firstly, bisimulation does not preserve the interpretation we would like to have for some constructs, and especially for internal choice. Moreover, given an encoding of a calculus into another, a bisimulation

semantics, being based on the interaction capabilities of the encoded terms, does not leave any room to play with the target calculus. On the other hand, in the may testing approach, one has the possibility to identify sub-calculi of the target, and see what is the resulting semantics on the source calculus. Let us explain these two points in more detail.

In investigating the full abstraction problem for the lazy λ -calculus, Abramsky found out that this calculus is too weak. To make it “complete” with respect to the canonical denotational semantics, one must enrich it with a convergence testing facility, and some parallel feature, see [2, 3]. Milner discovered that one can encode the *convergence testing combinator* C of Abramsky in the π -calculus, namely:

$$\mathcal{E}[\![C]\!]u =_{\text{def}} u(x, v)(\nu w)(\bar{x}w \mid \bar{w}xw. \mathcal{E}[\![\mathbf{I}]\!]v)$$

(note that one uses an output guard). This combinator is such that (CM) converges, to the identity \mathbf{I} , if and only if M converges. As for the parallel convergence testing combinator P , such that PMN converges, to the identity \mathbf{I} , if and only if M converges or N converges, one could encode it as follows, using output guards:

$$\mathcal{E}[\![P]\!]u =_{\text{def}} u(x, v)v(y, w)(\nu r)(\nu s)\bar{x}r.\bar{y}s.(\nu o)(\bar{r}xr.\bar{o} \mid \bar{s}ys.\bar{o} \mid o(\mathcal{E}[\![\mathbf{I}]\!]w))$$

However, parallel convergence testing is not the most general way to deal with parallel functions. In [5, 7] we have shown that parallel composition of functions is simply the *join* in the denotational semantics, which may be represented in the syntax by internal choice. For instance, $P = (\lambda xy.Cx) \oplus (\lambda xy.Cy)$. Obviously, to preserve some properties of the interpretation of internal choice as a join, like for instance $\mathbf{I} \oplus \Omega = \mathbf{I}$, one should replace the bisimulation by a weaker notion, e.g. a simulation semantics, as given by the Definition 2.3. But then it is not clear that adding internal choice to the λ -calculus makes it as powerful as the π -calculus.

Clearly, the may testing semantics is far less sensitive than bisimulation to the specific way we describe the operational behaviour of the constructs of a calculus. It is closer to the denotational approach, whose purpose is precisely to abstract away from this operational description. Moreover, the may testing scenario leaves some room for questions that could not be asked using a bisimulation semantics. For instance, one may ask whether it is possible to identify interesting sub-calculi of the π -calculus, making it closer to the “complete” lazy λ -calculus, enriched with convergence testing and some parallel facility. As Lavatelli shows in [12], such a calculus may be adequately encoded into the π -calculus, though the encoding is not fully abstract – and we have seen why: the π -calculus adds the possibility of creating and detecting deadlocks. Now, provided that we adopt the may testing semantics, the encoding of the λ -calculus is not very much affected by modifying the translation of abstractions, as follows:

$$\mathcal{E}[\![\lambda xM]\!]u = !u(x, v)\mathcal{E}[\![M]\!]v$$

The encoding then goes into a sub-calculus where any input prefix takes the form $!u(v_1, \dots, v_k)P$. Actually, since the law $!P \equiv (P \mid !P)$ is still useful, it would be more convenient to adopt a different notion of reduction, namely

$$(\bar{u}\tilde{w} \mid u(\tilde{v})P) \rightarrow ([\tilde{w}/\tilde{v}]P \mid u(\tilde{v})P)$$

rather than modifying the syntax. Then one could interpret $u(v_1, \dots, v_k)P$ as providing the “service” P under the name u , which does not disappear once requested, while a message $\bar{u}w_1 \cdots w_k$ is a call to this service, or an application of it, with parameters w_1, \dots, w_k . In this “ $\pi!$ -calculus” internal choice is still definable – more generally, $\pi!$ may be used as a target to encode the λ -calculus with “infinite resources” $\Lambda_{\infty r}$. Therefore one may conjecture that the bisimulation semantics induced by this encoding on λ -terms is still the equality of L evy-Longo trees. On the other hand, one may wonder whether the may testing semantics induced on λ -terms by this kind of restricted π -calculus (we should also allow output guards, to be able to define convergence testing) differs from the canonical denotational semantics.

References

- [1] M. Abadi, L. Cardelli, P. L. Curien, and J. J. L evy. Explicit substitutions. *Journal of Functional Programming*, 1:375 – 416, 1991.
- [2] S. Abramsky. The Lazy Lambda Calculus. In D. Turner, editor, *Research Topics in Functional Programming*, pages 65 – 116. Addison-Wesley, 1989.
- [3] S. Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Information and Computation*, 105(2):159 – 267, 1993.
- [4] H.P. Barendregt. *The Lambda Calculus*. North-Holland, 1985.
- [5] G. Boudol. A Lambda Calculus for Parallel Functions. Technical Report 1231, INRIA Sophia-Antipolis, May 1990.
- [6] G. Boudol. The lambda calculus with multiplicities. Technical Report 2025, INRIA Sophia-Antipolis, September 1993. Available as `ftp/pub/papers/boudol/lambdam-prelim.ps.Z` by anonymous ftp at `cma.cma.fr`.
- [7] G. Boudol. Lambda-calculi for (strict) parallel functions. *Information and Computation*, 108(1):51–127, 1994.
- [8] G. Boudol and C. Laneve. The discriminating power of multiplicities in the λ -calculus. Technical Report 2441, INRIA Sophia-Antipolis, December 1994.
- [9] G. Boudol and C. Laneve. Termination, deadlock and divergence in the λ -calculus with multiplicities. To appear in the Proceedings of the 11th Mathematical Foundations of Programming Semantics Conference, 1995.
- [10] U. Engberg and M. Nielsen. A calculus of communicating systems with label passing. Technical report, Daimi PB-208, Aarhus University, 1986.
- [11] R. Hindley and G. Longo. Lambda-calculus models and extensionality. *Zeitschr. f. Math. Logik und Grundlagen d. Math.*, 26:289–310, 1980.

-
- [12] C. Lavatelli. Non-deterministic lazy λ -calculus vs π -calculus. Technical report, LIENS-93-15, Ecole Normale Supérieure, Paris, 1993.
- [13] J.J. Lévy. An algebraic interpretation of the $\lambda\beta K$ -calculus; and an application of a labelled λ -calculus. *Theoretical Computer Science*, 2(1):97 – 114, 1976.
- [14] J.J. Lévy. *Réductions Correctes et Optimales dans le Lambda Calcul*. PhD thesis, Université Paris VII, 1978.
- [15] G. Longo. Set theoretical models of lambda calculus: Theories, expansions and isomorphisms. *Annals of Pure and Applied Logic*, 24:153 – 188, 1983.
- [16] R. Milner. Fully abstract models of typed λ -calculi. *Theoretical Computer Science*, 4:1 – 22, 1977.
- [17] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [18] R. Milner. Functions as processes. *Mathematical Structures in Computer Science*, 2:119 – 141, 1992.
- [19] R. Milner. The Polyadic π -Calculus: A Tutorial. In F. Bauer, W. Brauer, and H. Schwichtenberg, editors, *Logic and Algebra of Specification*, pages 203 – 246. Springer Verlag, 1993.
- [20] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. *Information and Computation*, 100(1):1 – 77, 1992.
- [21] C.-H. Luke Ong. *The Lazy Lambda Calculus: an Investigation into the Foundations of Functional Programming*. PhD thesis, Imperial College of Science and Technology, University of London, 1988.
- [22] G. Plotkin. Call-by-name, call-by-value and the λ -calculus. *Theoretical Computer Science*, 1:125 – 159, 1975.
- [23] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis, University of Edinburgh, 1993.
- [24] D. Sangiorgi. The lazy lambda calculus in a concurrency scenario. *Information and Computation*, 120(1):120 – 153, 1994.
- [25] D. Sangiorgi. Lazy functions and mobile processes. 1995.
- [26] C. Wadsworth. The relation between computational and denotational properties for Scott D_∞ -model of the lambda-calculus. *SIAM Journal on Computing*, 5:488 – 521, 1976.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399