Math. Struct. in Comp. Science (1999), *vol.* 9, *pp.* 437–482. Printed in the United Kingdom

A semantics for lambda calculi with resources

GÉRARD BOUDOL[†], PIERRE-LOUIS CURIEN[‡] and CAROLINA LAVATELLI[‡]

[†] *INRIA, BP93, F-06902 Sophia Antipolis Cedex*

[‡] *ENS DMI, 45 Rue d'Ulm, F-75230 Paris, Cedex 05*

Received 30 September 1998; revised 15 April 1999

We present the λ-calculus with resources λ_r, and two variants of it: a deterministic restriction λ_m and an extension λ_r^c with a convergence testing operator. These calculi provide a control on the substitution process – deadlocks may arise if not enough resources are available to carry out all the substitutions needed to pursue a computation. The design of these calculi was motivated by Milner's encoding of the λ-calculus in the π-calculus. As Boudol and Laneve have shown elsewhere, the discriminating power of λ_m (given by the contextual observational equivalence) over λ-terms coincides with that induced by Milner's π-encoding, and coincides also with that provided by the lazy algebraic semantics (Lévy–Longo trees). The main contribution of this paper is model-theoretic. We define and solve an appropriate domain equation, and show that the model thus obtained is fully abstract with respect to λ_r^c. The techniques used are in the line of those used by Abramsky for the lazy λ-calculus, the main departure being that the resource-consciousness of our calculi leads us to introduce a non-idempotent form of intersection types.

1. Introduction

In this paper, we address the problem of providing a denotational semantics for Boudol's resource-conscious λ-calculus (Boudol 1993), defined with the aim of reducing the gap between the π-calculus (Milner *et al.* 1989) and the lazy λ-calculus (Abramsky 1989).

The study of the connection between lazy λ-calculus and π-calculus was started with Milner's encoding of λ into π (hereafter simply called the encoding), given in Milner (1990). Milner's encoding is adequate in the framework of a contextual operational semantics, but not fully abstract. That is, π-contexts over encoded λ-terms are strictly more discriminating than λ-contexts over the original terms. It has been observed that there are π-agents that behave over the encoding as (parallel) convergence testing combinators, which are not definable in pure lazy λ-calculus. On the other hand, as Boudol emphasized (Boudol 1993), the encoding does not verify the equality $xx = x(\lambda y.xy)$, though it is valid in weakly extensional models[†] of (extensions of) the λ-calculus (with parallel functions, non-deterministic choice, (parallel) convergence testing) (Abramsky 1989; Abramsky and

[†] By weakly extensional, we mean that the equation $M = \lambda y.(My)$ is valid for every term M that has a value (with y not free in M).

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Ong 1993; Boudol 1990; Boudol 1991). This indicates that, even if such extended λ-calculi enjoy adequate encodings into π (see for instance the encoding of lazy λ-calculus with non-deterministic choice and convergence testing in Lavatelli (1996)), these encodings will not be fully abstract†.

We argue that the failure of the aforementioned equality $xx = x(\lambda y.xy)$ is due to the fact that the usual λ-calculi lack the possibility of raising deadlocks during evaluation. Indeed, one can build a π-context with only one available resource for x, say, the identity, such that the first occurrence of x can be substituted by the identity but the second one stands for a deadlocked term. In this context, the evaluation of the encoding of xx stops in a deadlocked term while that of $x(\lambda y.xy)$ ends in an abstraction (a value). This phenomenon is at the origin of the definition of the λ-calculus with resources λ_r, a refinement of pure lazy λ-calculus that allows us to control the availability of arguments. In this calculus, arguments are multisets of terms, called resources, of infinite or finite cardinality. Infinite homogeneous arguments correspond exactly to those of pure λ-calculus, and in this case, β-reductions can involve an arbitrary number of substitutions. The possibility of raising deadlocks is introduced through the finite arguments; indeed, a finite number of resources implies a limited number of substitutions during β-reductions, and there may be more relevant free occurrences of a variable than there are resources for them. The calculus is non-deterministic because, by definition, the substitution mechanism does not follow any strategy for fetching resources from arguments. An interesting sub-calculus of λ_r is its deterministic version, called λ-calculus with multiplicities λ_m, where arguments are multisets of a unique term. Indeed, most of the results for the encoding of lazy λ-calculus into π were shown by Boudol and Laneve in the framework of multiplicities (Boudol and Laneve 1994; Boudol and Laneve 1995a; Boudol and Laneve 1995b).

The resource calculi admit different 'may testing' observational semantics (Boudol 1993; Boudol and Laneve 1994; Boudol and Laneve 1995a; Boudol and Laneve 1995b). Two main scenarios have been considered: the standard one, which takes abstractions only as observables and does not distinguish between deadlock and divergence, verifies η-expansion, while the flat one, which allows us to observe both abstractions and deadlocks, does not. The question raised by Milner (Milner 1990),

what is the semantics induced upon λ-terms by encoding them into the π calculus?,

has been answered as follows by Boudol and Laneve: it is the flat semantics induced by contexts of the λ-calculus with multiplicities. Moreover, they show that this semantics over pure λ-terms coincides with Lévy's algebraic semantics (Lévy 1976), and that, in fact, non-determinism does not add any extra discriminating power. As far as semantic equality is concerned, these results also hold in the standard scenario. The flat and the standard preorders correspond to two natural orderings on Lévy–Longo trees: the standard inclusion, and the Plotkin–Scott–Engeler ordering, which is essentially an η-extension of the first.

† Sangiorgi does achieve full abstraction for the encoding of an extension of the lazy λ-calculus with some form of non-determinism, but his result refers to a more powerful observational semantics (extended applicative bisimulation), which is intensional enough to distinguish $\lambda y.xy$ from x.

These results allow us to compare the expressiveness of λ_m (hence of λ_r) with that of classical extensions of lazy λ-calculus. Ong shows in Ong (1988) that the Plotkin–Scott–Engeler preorder over pure λ-terms is strictly more discriminating than the semantics induced by λ-contexts augmented with parallel convergence testing. Hence λ_m-contexts are strictly more discriminating than contexts of the lazy λ-calculus augmented with parallel convergence testing; in particular, neither a non-deterministic feature nor a convergence testing facility are needed to recover the power of π-contexts over pure λ-terms. However, if we consider more widely terms with resources instead of pure λ-terms, convergence testing does separate some λ_r-terms that λ_r-contexts cannot distinguish.

The present work concerns the denotational semantics of resource calculi. We propose a domain equation for these calculi whose canonical solution has a logical presentation based on a refinement of the intersection type discipline (Sallé 1978; Coppo and Dezani 1980; Coppo *et al.* 1980; Coppo *et al.* 1981). In systems with intersection, types are preserved by expansion. The crucial point is to group type information through the use of a conjunction: if $M[N/x]$ has type ϕ and $\sigma_1, \ldots, \sigma_n$ are the types given to N all along this derivation, then M has type ϕ under the assumption that x has type $\sigma_1 \wedge \cdots \wedge \sigma_n$. In agreement with the permanent availability of resources in classical λ-calculus, conjunction is idempotent, that is, $\phi \wedge \phi$ means exactly the same as ϕ. This is no longer suitable in resource calculi; the refinement, originally proposed in Boudol (1993), consists precisely in the elimination of the rule for contracting hypotheses. This type system induces an adequate semantics for λ_r with respect to the standard contextual or testing preorder. In Boudol and Lavatelli (1996), a counter-example to full abstraction is constructed and full abstraction of the type semantics is shown for λ_r^c, which is λ_r augmented with convergence testing.

The domain equation introduced in this paper has the form $D = (\mathscr{M}(D) \rightarrow D)_\perp$, where $\mathscr{M}(D)$ stands for the 'domain of multisets' of terms. From the strict point of view of semanticists, the equation is not completely satisfactory since the structure of $\mathscr{M}(D)$ does not allow for a direct interpretation of arguments: in fact, we interpret the terms MP rather than just the argument P, making use of all finite approximations of P. However, the fact that the canonical model induced by the equation is the exact counterpart of a simple and appropriate type system for our calculus justifies its study in our opinion. The tight relation with this economical type system is the key to our proof of full abstraction, as in the work of Abramsky and Ong (Abramsky and Ong 1993)

In Section 2, we deal with syntactic issues: we present the λ-calculi with resources, give an operational semantics, and discuss expressiveness issues. We show, in particular, that λ_r^c allows for more discriminations than λ_r on terms with multiplicities. The domain equation and the interpretation function are presented in Section 3. In Sections 4 and 5, we recall Boudol's type system and relate it to the domain introduced in Section 3: the meaning of a term coincides with the collection of its possible types. Finally, in Section 6, we prove that the model is fully abstract with respect to λ_r^c, while it is only adequate with respect to λ_r.

2. λ-calculi with resources

In pure λ-calculus, arguments are managed as permanent resources: β-conversion of $(\lambda x.M)N$ gives $M[N/x]$, a term where the argument N replaces x as many times as

x occurs free in M. In other words, β-conversion transforms the argument N in an inexhaustible resource of the substitution process.

The λ-calculi with resources allow us to model non-permanent as well as permanent resources. In these calculi, infinitely available arguments come with an explicit infinite multiplicity: standard application reads MN^∞. The limited availability of arguments, hence the possibility of reaching a deadlock during evaluation, and non-determinism follow from the introduction of finite bags or multi-sets of terms as arguments. Bags are of the form $(N_1^{m_1} \mid \cdots \mid N_k^{m_k})$ with $m_i \in \mathbb{N} \cup \{\infty\}$, where each component N_i is used as argument at most m_i times in a substitution process. The effective substitution of x by a multiset of cardinality q obeys two laws:

(1) there are at most q replacements, and

(2) these replacements are performed by necessity, that is, when x occurs in head position.

Explicit substitutions $\langle P/x \rangle$ in the style of Abadi *et al.* (1991) appear as a suitable computational device to model this kind of partial substitution, while keeping the lazy regime of evaluation. The following is an example of evaluation in resource calculi: assume $P_1, \ldots, P_n$ are bags of terms and let $Q = (N_1^{m_1} \mid \cdots \mid N_k^{m_k})$; then

$$xP_1 \cdots P_i\langle Q/x\rangle P_{i+1} \cdots P_n \;\to_r\; N_j P_1 \cdots P_i \langle Q'/x\rangle P_{i+1} \cdots P_n$$

for any $j \in \{1, \ldots, k\}$ where $Q' = (N_1^{m_1} \mid \cdots \mid N_j^{m_j - 1} \mid \cdots N_k^{m_k})$. It is worth noticing that the evaluation is non-deterministic since, above, any component may be fetched from Q. Notice also that the P_i's are left untouched by the reduction: the rest Q' of the bag is left available for when some other free occurrences of x will eventually reach a head position. Also, we assume as a side condition that x does not occur free in N_j, so that Q' will not concern N_j.

2.1. *Syntax*

In this section we define the three calculi used throughout the paper: λ-calculus with multiplicities λ_m and λ-calculus with resources λ_r, both defined in Boudol (1993), and λ-calculus with resources and convergence testing λ_r^c (Lavatelli 1996; Boudol and Lavatelli 1996).

Terms of λ_r^c are either variables taken from a countable set Var ranged over by u, v, w, x, y, $z, \ldots$, or abstractions $\lambda x.M$, or applications MP where P is the argument (a bag of terms), or expressions like $M\langle P/x\rangle$ where $\langle P/x\rangle$ is a substitution entry for M, or expressions $\mathsf{c}P$ for testing the convergence of bag P. The grammar is as follows:

$$\begin{array}{lll} \text{Terms}: & (\Lambda_{rc}) & M ::= x \mid \lambda x.M \mid (MP) \mid M\langle P/x\rangle \mid \mathsf{c}P \\ \text{Bags}: & (\Pi) & P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^\infty \,. \end{array}$$

The terms of λ_r are just those of λ_r^c not containing c. As a further restriction, the definition of λ_m allows homogeneous bags only as arguments, with finite or infinite multiplicity. That is, the grammar for λ_m-terms is as follows:

$$(\Lambda_m) \quad M ::= x \mid \lambda x.M \mid (MN^k) \mid M\langle N^k/x\rangle \quad \text{where } k \in \mathbb{N} \cup \{\infty\}\,.$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

$$
\begin{aligned}
y[z/x] &= \begin{cases} z & \text{if } y = x \\ y & \text{otherwise} \end{cases} \\
(\lambda y.M)[z/x] &= \begin{cases} \lambda y.M & \text{if } y = x \\ \lambda y'.M[y'/y][z/x] & \text{otherwise, } y' \text{ new} \end{cases} \\
(MP)[z/x] &= (M[z/x])(P[z/x]) \\
(M\langle P/y\rangle)[z/x] &= \begin{cases} M\langle P[z/x]/y\rangle & \text{if } y = x \\ (M[z/x])\langle P[z/x]/y\rangle & \text{otherwise} \end{cases} \\
(\mathrm{c}P)[z/x] &= \mathrm{c}(P[z/x]) \\
\mathbf{1}[z/x] &= \mathbf{1} \\
(P \mid Q)[z/x] &= (P[z/x] \mid Q[z/x]) \\
(M^\infty)[z/x] &= (M[z/x])^\infty
\end{aligned}
$$

Fig. 1. Renaming

Here, the argument N^0 stands for $\mathbf{1}$, and N^k with finite $k > 0$ stands for the bag $(\underbrace{N \mid \cdots \mid N}_{k \text{ times}})$. We shall omit multiplicity 1 in arguments, that is, MN will be used for MN^1.

We adopt the convention that $L, M, N, \ldots$ and $P, Q, \ldots$ denote terms and bags, respectively. Moreover, T stands indistinctly for bags or terms, and $R, S \ldots$ represent either arguments or substitution entries. We use $\tilde{R}$ as a short form for the sequence $R_1 \ldots R_n$, when n is not relevant or is known from the context. For sequences composed of substitution entries only, we often write $\langle \tilde{P}/\tilde{x}\rangle$ instead of $\langle P_1/x_1\rangle \cdots \langle P_n/x_n\rangle$.

Free variables of terms are defined as usual, with the following additions:

$$
\begin{gathered}
fv(M\langle P/x\rangle) = fv(M) \setminus \{x\} \cup fv(P) \quad fv(\mathrm{c}P) = fv(P) \\
fv(\mathbf{1}) = \varnothing \quad fv(P \mid Q) = fv(P) \cup fv(Q) \quad fv(M^\infty) = fv(M)\,.
\end{gathered}
$$

Some special terms will be used throughout the paper. They are

$$
\mathbf{I} = \lambda x.x \qquad \Omega = (\lambda x.xx^\infty)(\lambda x.xx^\infty)^\infty\,.
$$

We consider λ_r^c-terms up to α-conversion, whose definition involves the renaming operation given in Figure 1. The α-conversion $M{=_\alpha}N$ is the congruence generated by the following laws, when $z \notin var(M)$: $\lambda x.M = \lambda z.M[z/x]$ and $M\langle P/x\rangle = (M[z/x])\langle P/z\rangle$.

The congruence $\equiv$, called structural equivalence, defined in Figure 2 allows us to permute the resources of a bag and to develop infinite multiplicities.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

$$M^\infty \equiv (M \mid M^\infty) \qquad (P \mid Q) \equiv (Q \mid P)$$

$$P \equiv (\mathbf{1} \mid P) \qquad (P \mid (Q \mid T)) \equiv ((P \mid Q) \mid T)$$

$$P \equiv Q \Rightarrow \begin{cases} MP \equiv MQ \\ M\langle P/x\rangle \equiv M\langle Q/x\rangle \end{cases}$$

Fig. 2. Structural equivalence

2.2. *Evaluation*

The evaluation of λ_r^c-terms follows the lazy strategy adopted by Abramsky and Ong for the λ-calculus: neither the body of abstractions nor the arguments in application terms are evaluated. Moreover, the convergence testing combinator introduced by Abramsky and Ong (Abramsky and Ong 1993) can be defined as $\lambda x.cx$. The set of evaluation rules consists of two parts and is given in Figures 4 and 3, respectively. The first part formalizes weak β-reduction $\rightarrow_{rc}$ using explicit substitutions: there are five axioms and four structural rules. The second part establishes a mechanism for *fetching* resources, allowing us to perform substitutions in a delayed manner through an auxiliary relation $\succ$. As usual, $\rightarrow_{rc}^{\star}$ stands for zero or more evaluation steps, $\rightarrow_{rc}^{+}$ stands for one or more evaluation steps.

The axiom for *fetch*, combined with the fact that we have defined evaluation up to the structural equivalence, allows us to encode non-deterministic choice at the level of terms (Boudol 1993). Indeed, setting $(M{\oplus}N) \stackrel{def}{=} x\langle (M \mid N)/x\rangle$ and using the fetch rule, we have both

$$(M{\oplus}N) \rightarrow_{rc} M\langle N/x\rangle \qquad \text{and} \qquad (M{\oplus}N) \rightarrow_{rc} N\langle M/x\rangle,$$

provided that $x \notin fv(M) \cup fv(N)$. The resulting terms $M\langle N/x\rangle$ and $N\langle M/x\rangle$ are essentially M and N, respectively (see Lemma 2.5).

$$x\langle N/x\rangle \succ N$$

$$M\langle N/x\rangle \succ M' \Rightarrow \begin{cases} MP\langle N/x\rangle \succ M'P \\ (\mathrm{c}M)\langle N/x\rangle \succ \mathrm{c}M' \\ M\langle P/z\rangle\langle N/x\rangle \succ M'\langle P/z\rangle \quad \text{if } x \neq z \text{ and } z \notin fv(N) \end{cases}$$

Fig. 3. Fetching rules

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

$(\beta)\ (\lambda x.M)P \ \rightarrow_{rc}\ M\langle P/x\rangle$

$(v)\ (\lambda x.M)\langle P/z\rangle \ \rightarrow_{rc}\ \lambda x.(M\langle P/z\rangle)\quad \text{if } x \notin fv(z)\cup fv(P)$

$(\text{c1})\ \mathsf{c}(\lambda x.M)\ \rightarrow_{rc}\ \mathbf{I}$

$(\text{c2})\ \mathsf{c}P\ \rightarrow_{rc}\ \mathsf{c}M\quad \text{if } P \equiv (M \mid Q)$

$(fetch)\ M\langle N/x\rangle{\succ}M' \ \Rightarrow\ M\langle (N\mid Q)/x\rangle \ \rightarrow_{rc}\ M'\langle Q/x\rangle\quad \text{if } x \notin fv(N)$

$$M \ \rightarrow_{rc}\ M' \ \Rightarrow \begin{cases} MP \ \rightarrow_{rc}\ M'P & \\ M\langle P/x\rangle \ \rightarrow_{rc}\ M'\langle P/x\rangle & \\ \mathsf{c}M \ \rightarrow_{rc}\ \mathsf{c}M' & \\ N \ \rightarrow_{rc}\ M' & \text{if } M \equiv N \end{cases}$$

Fig. 4. Evaluation in λ_r^c

The first rule for c establishes that the testing is successful when a value is encountered; the whole term becomes the identity so that evaluation can be pursued. Here is an illustration: $(\mathsf{c}(\lambda x.M))(N \mid Q) \ \rightarrow_{rc}\ \mathbf{I}(N \mid Q) \ \rightarrow_{rc}^{\star}\ N\langle Q/x\rangle$ (with $x \notin fv(Q)$). The second rule selects non-deterministically a component M from P and discards the rest of the bag. The term M becomes the argument of c. The structural rule for c allows us to evaluate its argument.

We next illustrate the fetch operation. If $M = xR_1 \cdots R_n\langle N/x\rangle$, then a derivation $M{\succ}M'$ has the following shape:

$$\cfrac{\cfrac{\cfrac{x\langle N/x\rangle{\succ}N}{xR_1\langle N/x\rangle{\succ}NR_1}}{\vdots}}{\cfrac{xR_1\cdots R_{n-1}\langle N/x\rangle{\succ}NR_1\cdots R_{n-1}}{M = xR_1\cdots R_n\langle N/x\rangle{\succ}M' = NR_1\cdots R_n\,.}}$$

The rule involving c allows us to look for a substitution entry for x if the term M being tested has x as head variable.

The evaluation relations $\rightarrow_r$ and $\rightarrow_m$ for λ_r and λ_m, respectively, are defined by the subset of rules defining $\rightarrow_{rc}$ not dealing with c. Notice that for $\rightarrow_m$ the fetch actually means

$$\frac{M\langle N/x\rangle{\succ}M'\ \ (x \notin fv(N))}{M\langle N^{k+1}/x\rangle \ \rightarrow_m\ M'\langle N^k/x\rangle}$$

where we consider $k = k + 1$ if $k = \infty$, and hence it is deterministic. Also, Rule (c2) boils down to $\mathrm{c}M^k \rightarrow_{rc} \mathrm{c}M$ (for $k \geqslant 1$). Thus, unlike λ_r and λ_r^c, the calculus with multiplicities λ_m is deterministic up to α-conversion.

2.3. *Operational semantics*

We observed in the introduction that a terminating evaluation of a λ_r^c-term may end in an abstraction or a deadlock. Indeed, terms like $xP_1 \cdots P_n\langle \mathbf{1}/x\rangle$ are legal and may be reached during evaluation. Nevertheless, the operational semantics adopted here for λ_r^c takes abstractions only as values and does not provide any mean to detect deadlocks. Hereafter, V, W range over values, that is, over abstractions. The *convergence predicate* $\Downarrow_{rc}^l$ is defined on closed terms by

$$M \Downarrow_{rc}^l V \quad \stackrel{def}{\Longleftrightarrow} \quad (M \rightarrow_{rc}^\star V \text{ in } l \text{ steps and } V \text{ is an abstraction})\,. \tag{1}$$

That is, a term M converges whenever at least one evaluation issued from it ends in a value. We use the notations $M \Downarrow_{rc}^l$ and $M{\Downarrow_{rc}}$ as short forms of $(\exists V\ M \Downarrow_{rc}^l V)$. We say that M diverges whenever it does not converge, written $M{\Uparrow_{rc}}$. The convergence and divergence predicates for λ_r and λ_m are defined similarly; we denote them by $\Downarrow_r$, $\Uparrow_r$ and $\Downarrow_m$, $\Uparrow_m$, respectively.

We adopt as observational semantics the extensional preorder of Morris, also called *testing preorder*, where a term is approximated by another if it passes at least as many tests as the first one. The tests for terms are made up of the constructors of the language plus a constant [] (pronounced 'hole'). Tests are usually called contexts and are ranged over by capital letters A, B, C, D. We let $C[M]$ denote the term of Λ_{rc} obtained by replacing in C all occurrences of [] by M. As a result of these replacements, free variables of M may become bound in $C[M]$. We say that C closes M if all free variables of M are bound in $C[M]$. A term M passes a test C whenever $C[M]$ converges.

Definition 2.1. (Observational semantics) The testing preorder $\sqsubseteq_{rc}$ is defined as follows:

$$(M \sqsubseteq_{rc} N) \stackrel{def}{\Longleftrightarrow} (C[M]{\Downarrow_{rc}} \Rightarrow C[N]{\Downarrow_{rc}} \text{ for all contexts } C \text{ closing } M, N)\,.$$

The testing preorders for λ_r and λ_m, $\sqsubseteq_r$ and $\sqsubseteq_m$, respectively, have similar definitions. The testing preorder is also called the observational preorder.

The fact that contexts used for testing terms are of arbitrary kind makes the definition of the preorder $\sqsubseteq_{rc}$ unworkable, although it is a precongruence, that is, if $M \sqsubseteq_{rc} N$ then $C[M] \sqsubseteq_{rc} C[N]$ for any context C. But it is possible to give an alternative presentation of $\sqsubseteq_{rc}$ in terms of a restricted set of *applicative contexts*. Applicative contexts contain at most one hole, placed in head position; their syntax is given by the following grammar:

$$A ::= [\,] \mid AP \mid A\langle P/x\rangle \mid \mathrm{c}A\,.$$

Definition 2.2. The applicative preorder $\sqsubseteq_{\mathscr{A}}$, with associated equivalence $\simeq_{\mathscr{A}}$, is defined by

$$(M \sqsubseteq_{\mathscr{A}} N) \stackrel{def}{\Longleftrightarrow} (A[M]{\Downarrow_{rc}} \Rightarrow A[N]{\Downarrow_{rc}} \text{ for all applicative contexts } A \text{ closing } M, N)\,.$$

Lemma 2.3. (Context lemma) For all $M, N \in \Lambda_{rc}$,

$$(M \sqsubseteq_{rc} N) \iff (M \sqsubseteq_{\mathscr{A}} N)\,.$$

Proof. A detailed proof can be found in Lavatelli (1996), a short one in Boudol and Lavatelli (1996). □

The evaluation relation is decreasing with respect to the applicative preorder.

Lemma 2.4. $(M \to_{rc} N) \Rightarrow (N \sqsubseteq_{\mathscr{A}} M)$.

Proof. Assume $M \to_{rc} N$ and $A[N]\Downarrow_{rc} V$ for some abstraction V. Since the term M appears at the head position in $A[M]$, we have $A[M] \to_{rc} A[N]$, and hence $A[M]\Downarrow_{rc} V$. □

Clearly, some permutations of substitution entries or garbage collection may be done in a term without affecting its computational content. To state this formally, let $\asymp$ be the least relation containing $\equiv \cup =_\alpha$ and satisfying

$$\begin{array}{rcll} M\langle R/x\rangle & \asymp & M & x \notin fv(M) \\ (MP)\langle R/x\rangle & \asymp & (M\langle R/x\rangle)P & x \notin fv(P) \\ M\langle P/z\rangle\langle R/x\rangle & \asymp & M\langle R/x\rangle\langle P/z\rangle & z \neq x, x \notin fv(P), z \notin fv(R) \\ (\mathsf{c}M)\langle R/x\rangle & \asymp & \mathsf{c}(M\langle R/x\rangle) & \end{array}$$

$$\begin{array}{rcl} M \asymp M' & \Rightarrow & (MP) \asymp (M'P) \\ M \asymp M' & \Rightarrow & (M\langle P/x\rangle) \asymp (M'\langle P/x\rangle) \\ M \asymp M' & \Rightarrow & (\mathsf{c}M) \asymp (\mathsf{c}M')\,. \end{array}$$

Notice that the last three implications are equivalent to stating that if $M \asymp M'$, then $A[M] \asymp A[N]$ for any applicative context A.

Lemma 2.5. $(M \asymp N) \Rightarrow (M \simeq_{\mathscr{A}} N)$.

Proof. The lemma is a consequence of the following easy property (Lavatelli 1996):

$$(*)\ \ (M \asymp N \ \text{ and } \ M \to_{rc} M') \Rightarrow \exists N'\ (N \to_{rc} N' \ \text{ and } \ M' \asymp N')\,.$$

Let $M \asymp N$ and A be an applicative context, so that $A[M] \asymp A[N]$. Suppose $A[M]\Downarrow_{rc} V$. Then by $(*)$, there is some N' such that $A[N] \to_{rc}^{\star} N' \asymp V$. Given that V is an abstraction, the only possible case of the definition of $\asymp$ that can apply is the first one. Thus $N' = V\langle R/x\rangle$ with $x \notin fv(V)$. But then, setting $V = \lambda y.N_1$, we have $N' \to_{rc} \lambda y.N_1\langle R/x\rangle$, and hence $A[N]\Downarrow_{rc}$. □

From now on, substitution entries on variables that do not occur free will often be discarded for the sake of clarity (notably in the examples of Section 2.4).

The following two technical lemmas will be used in the proofs of Lemmas 6.14 and 6.8, respectively.

Lemma 2.6. $(MP \simeq_{\mathscr{A}} My^{\infty}\langle P/y\rangle)$, for $y \notin fv(M) \cup fv(P)$.

Proof. If M is an abstraction, the proof relies on the following auxiliary equivalence

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

(with $y \notin fv(M)$):

$$(*)\quad M\langle P/x\rangle \simeq_{\mathscr{A}} M\langle y^{\infty}/x\rangle\langle P/y\rangle\ .$$

This property is proved by tedious case inspection and induction on the length of the evaluation. The entry $\langle y^{\infty}/x\rangle$ behaves as a buffer: each use of a resource of P on the right-hand side is simulated by a use of y (which is always available due to the infinite multiplicity) immediately followed by a use of the same resource of P on the left-hand side, and conversely. The detailed proof can be found in Lavatelli (1996). □

Lemma 2.7. Let $\langle P_0/x\rangle, \langle P_1/x\rangle, \ldots, \langle P_n/x\rangle$ be closed substitution entries on the same variable, and assume $P \equiv (N_1 \mid \cdots \mid N_n \mid Q)$. Then

$$(M\langle P_0/x\rangle)(N_1\langle P_1/x\rangle \mid \cdots \mid N_n\langle P_n/x\rangle \mid Q) \sqsubseteq_{\mathscr{A}} (MP)\langle (P_0 \mid P_1 \mid \cdots \mid P_n)/x\rangle\ .$$

Proof. The statement is shown together with the following two properties:

$$M\langle P_1/x\rangle\tilde{S}_0\langle P_2/x\rangle\tilde{S}_1 \sqsubseteq_{\mathscr{A}} M\tilde{S}_0\langle (P_1 \mid P_2)/x\rangle\tilde{S}_1$$

$$M\langle (N_1\langle P_1/x\rangle \mid \cdots \mid N_n\langle P_n/x\rangle \mid Q)/z\rangle\langle P_0/x\rangle \sqsubseteq_{\mathscr{A}} M\langle P/z\rangle\langle (P_0 \mid P_1 \mid \cdots \mid P_n)/x\rangle\ .$$ □

2.4. *Expressiveness*

Boudol and Laneve established the following result (Boudol and Laneve 1994): λ_m-contexts over pure λ-terms are strictly more discriminating than λ-contexts augmented with the parallel convergence testing combinator p (Abramsky 1989; Abramsky and Ong 1993), whose behaviour is specified as follows:

$$\begin{cases} \mathrm{p}MN \to \mathbf{I} & \text{if either } M \text{ or } N \text{ converges} \\ \mathrm{p}MN \text{ diverges} & \text{otherwise.} \end{cases}$$

(Notice that the convergence testing operator is recovered from the parallel convergence testing by setting $\mathrm{c} = \lambda x.\mathrm{p}xx$.) This is a fairly powerful result since c is not definable within λ_r (and hence neither is p); this follows from Example 2.12, where we exhibit two terms with multiplicities that are not separable by λ_r-contexts, but can be separated using a λ_r^c-context.

Examples 2.10 and 2.11 illustrate how multiplicities allow us to separate λ-terms that are naturally discriminated using convergence testing (also called sequential convergence testing) and parallel convergence testing, respectively, while Example 2.8 shows that the increase of expressivity is in fact strict.

Example 2.8. Let $M = xx^{\infty}$, $N = x(\lambda y.xy^{\infty})^{\infty}$. These (pure) λ-terms are equal in the theory induced by the applicative bisimulation (Abramsky 1989), even extended with parallel convergence testing combinators, but separable by means of resource contexts. Assume $C = [\,]\langle \mathbf{I}/x\rangle$ (giving multiplicity 1 to $\mathbf{I}$). Then

$$C[N] \to_m \mathbf{I}(\lambda y.xy^{\infty})^{\infty}\langle \mathbf{1}/x\rangle \to_r z\langle (\lambda y.xy^{\infty})^{\infty}/z\rangle\langle \mathbf{1}/x\rangle \to_r^{\star} \lambda y.(xy^{\infty}\langle \mathbf{1}/x\rangle) \Downarrow_m$$

$$C[M] \to_m \mathbf{I}x^{\infty}\langle \mathbf{1}/x\rangle \to_r z\langle x^{\infty}/z\rangle\langle \mathbf{1}/x\rangle \to_r x\langle x^{\infty}/z\rangle\langle \mathbf{1}/x\rangle \asymp x\langle \mathbf{1}/x\rangle \Uparrow_m$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

where the last evaluation diverges since $x\langle \mathbf{1}/x\rangle$ is a deadlocked term. Notice that here the ability to stop evaluation does not depend on the non-deterministic features of some of the resource calculi.

Remark 2.9. There are other ways to distinguish terms like xx and $x(\lambda y.xy)$, for example:

- A discipline for controlling the arity of functions rather than the uses of their arguments has been investigated independently in Piperno (1995) and in Curien (1998) and Curien and Herbelin (1998)†. This control ensures that a substitution, say, by a function of one argument can only be carried out on those occurrences that have explicitly one argument. Under this discipline we have $(xx)[\mathbf{I}/x] \rightarrow (\mathbf{I}x)[\mathbf{I}/x] \rightarrow x[\mathbf{I}/x]$ where the last term is deadlocked. On the other hand, $(x(\lambda y.xy))[\mathbf{I}/x]$ converges: $(x(\lambda y.xy))[\mathbf{I}/x] \rightarrow^\star (\lambda y.xy)[\mathbf{I}/x] \rightarrow \lambda y.y$. As far as we know, the semantic aspects of arity control have not been investigated.
- The equality of Böhm trees has been characterized in Dezani *et al.* (1998) observationally in an extension of the (non-lazy) λ-calculus with a non-deterministic choice operator and constants for numerals.

Example 2.10. Define $B = x(\lambda y.\Omega)\Omega$ and let

$$M = \lambda x.xB(\lambda y.\Omega) \quad \text{and} \quad N = \lambda x.x(\lambda z.Bz)(\lambda y.\Omega).$$

Abramsky and Ong (Abramsky and Ong 1993) used these terms to show that the convergence testing combinator adds some separation power to ordinary λ-contexts. Indeed, the context $A = [\,]\mathrm{c}$ is such that $A[N]$ converges and $A[M]$ diverges:

$$\begin{array}{l} A[N] \rightarrow \mathrm{c}(\lambda z.\mathrm{c}(\lambda y.\Omega)\Omega z)(\lambda y.\Omega) \rightarrow \mathbf{I}(\lambda y.\Omega) \rightarrow \lambda y.\Omega \\ A[M] \rightarrow \mathrm{c}(\mathrm{c}(\lambda y.\Omega)\Omega)(\lambda y.\Omega) \rightarrow \mathrm{c}(I\Omega)(\lambda y.\Omega) \rightarrow \mathrm{c}\Omega(\lambda y.\Omega)\,. \end{array}$$

In λ_m, M and N are separated by the context $C = [\,]U^1$, where $U = \lambda vw.v$:

$$\begin{array}{lll} C[N] & \rightarrow_m & x(\lambda z.Bz)(\lambda y.\Omega)\langle U/x\rangle \\ & \rightarrow_m & (\lambda vw.v)(\lambda z.Bz)(\lambda y.\Omega)\langle \mathbf{1}/x\rangle \\ & \rightarrow_m^\star & v\langle \lambda z.Bz/v\rangle\langle \lambda y.\Omega/w\rangle\langle \mathbf{1}/x\rangle \\ & \rightarrow_m^\star & \lambda z.(Bz\langle \mathbf{1}/v\rangle\langle \lambda y.\Omega/w\rangle\langle \mathbf{1}/x\rangle) \Downarrow_m \\ & & \\ C[M] & \rightarrow_m^\star & v\langle B/v\rangle\langle \lambda y.\Omega/w\rangle\langle \mathbf{1}/x\rangle \\ & \rightarrow_m & x(\lambda y.\Omega)\Omega\langle \mathbf{1}/v\rangle\langle \lambda y.\Omega/w\rangle\langle \mathbf{1}/x\rangle \Uparrow_m \;. \end{array}$$

Notice that the term $C[M]$ diverges just because the argument U can be used just once, that is, no resource is available for the second occurrence of x.

Example 2.11. Define $B = x\Omega\Omega$ and let

$$M = \lambda x.xB(xB\Omega) \quad \text{and} \quad N = \lambda x.xB(x(\lambda y.By)\Omega).$$

† Piperno defines a restriction of the β-reduction, while Curien and Herbelin consider a stack-free abstract machine that gives rise to a natural game-theoretic interpretation. In both frameworks a type-free strong normalization result holds.

These terms, taken from Boudol and Laneve (1994), illustrate the separation power of parallel convergence testing over pure λ-terms: M and N are not separable by contexts with sequential convergence testing, but can be separated if parallel convergence testing is allowed. We first explain informally why M and N are not separable in the lazy λ-calculus augmented with c. Below, the divergence $\Uparrow_\mathrm{C}$, the convergence $\Downarrow_\mathrm{C}$, and the observational equivalence $\simeq_\mathrm{C}$ are relative to this language. Assume $A = [\,]LL_1 \cdots L_n$, then the observational equivalence of M and N follows from the following two facts:

1. $(L\Omega\Omega) \Uparrow_\mathrm{C} \Rightarrow\ A[M] \Uparrow_\mathrm{C}$ and $A[N] \Uparrow_\mathrm{C}$. We examine the different reasons for the divergence of $L\Omega\Omega$:
   - L diverges, and then $A[M]$ and $A[N]$ diverge too; or
   - L takes one argument that is put into head position so that the statement holds, since both $A[M]$ and $A[N]$ begin with $L(L\Omega\Omega)$; or
   - L takes two arguments and puts one of them in head position: if it is the first one, then both $A[M]$ and $A[N]$ diverge since $(L\Omega\Omega) \Uparrow_\mathrm{C}$; if it is the second one, the statement holds, since then $(L(L\Omega\Omega)\Omega) \Uparrow_\mathrm{C}$ and $(L(\lambda y.(L\Omega\Omega)y)\Omega) \Uparrow_\mathrm{C}$.
2. $(L\Omega\Omega) \Downarrow_\mathrm{C} \Rightarrow\ A[M] \Downarrow_\mathrm{C}$ and $A[N] \Downarrow_\mathrm{C}$. This holds, since for any closed H we have $(H \Downarrow_\mathrm{C}) \Rightarrow (H \simeq_\mathrm{C} \lambda y.Hy)$. Therefore $ML \simeq_\mathrm{C} NL$ holds, and this implies $A[M] \simeq_\mathrm{C} A[N]$.

Fact (2) still holds for a λ_P-context. But (1) does not. Let $A = [\,]\mathrm{p}$, then we have

$$A[N] \to \mathrm{p}(\mathrm{p}\Omega\Omega)\underbrace{(\mathrm{p}(\lambda x.(\mathrm{p}\Omega\Omega)x)\Omega)}_{\text{converges}} \Downarrow_\mathrm{P} \qquad A[M] \to \mathrm{p}(\mathrm{p}\Omega\Omega)\underbrace{(\mathrm{p}(\mathrm{p}\Omega\Omega)\Omega)}_{\text{diverges}} \Uparrow_\mathrm{P} .$$

Notice that the convergence of $A[N]$ could be achieved thanks to the ability of p to choose its second argument in $\mathrm{p}(\mathrm{p}\Omega\Omega)(\mathrm{p}(\lambda x.(\mathrm{p}\Omega\Omega)x)\Omega)$, and then its first argument in $\mathrm{p}(\lambda x.(\mathrm{p}\Omega\Omega)x)\Omega$. In the framework of multiplicities, Boudol and Laneve show how Böhm's technique (Barendregt 1984; Krivine 1991) can be used to separate these two terms. Let $C = [\,]P^2FK$, where $P = \lambda x_1x_2x_3.x_3x_1x_2$, $K = \lambda x_1x_2.x_1$ and $F = \lambda x_1x_2.x_2$. The evaluation of $C[N]$ converges with two uses of P, while $C[M]$ would need three. During the evaluation of $C[N]$ and $C[M]$, the first use of P consumes F, bringing the subterms $x(\lambda z.Bz)\Omega$ and $xB\Omega$, respectively, into head position. Then P is used once again and consumes K, bringing $\lambda z.Bz$ and $B = x\Omega\Omega$, respectively, into head position. That is, the evaluation of $C[N]$ ends in an abstraction, while that of $C[M]$ ends in a deadlocked term:

$$\begin{array}{lll}
C[N] & \to_m^\star & x_3x_1x_2\langle B/x_1\rangle\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle\langle F/x_3\rangle K \\
 & \to_m & Fx_1x_2\langle B/x_1\rangle\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle K \\
 & \to_m^\star & x_2\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle K \\
 & \to_m & x(\lambda y.By)\Omega\langle P/x\rangle K \\
 & \to_m & P(\lambda y.By)\Omega\langle \mathbf{1}/x\rangle K \\
 & \to_m^\star & Kx_1x_2\langle \lambda y.By/x_1\rangle\langle \Omega/x_2\rangle\langle \mathbf{1}/x\rangle \\
 & \to_m^\star & (\lambda y.By)\langle \mathbf{1}/x\rangle \\
 & \to_m & \lambda y.(By\langle \mathbf{1}/x\rangle) \Downarrow_m \\
 & & \\
C[M] & \to_m^\star & B\langle \mathbf{1}/x\rangle = x\Omega\Omega\langle \mathbf{1}/x\rangle \ \Uparrow_m \quad .
\end{array}$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Example 2.12. Define $L = (\lambda y.\Omega)$ and $B = \lambda zx.xz^\infty$, and let

$$M = BL^m \quad \text{and} \quad N = BL^n \quad \text{with} \quad n > m \geqslant 1.$$

The terms M and N are not separable within λ_r: at most one resource L from the argument can be used during an evaluation (a proof of this can be found in Lavatelli (1996)). In fact, both M and N behave as BL^1 in any context. Let us illustrate this by an example: let $C = [\,](\mathbf{I}|Q_1)Q_2 \cdots Q_p$ with $p \geqslant 1$ and $T = BL^k$, with $k \geqslant 1$; then

$$\begin{array}{lll} C[T] & \to_r & (\lambda x.xz^\infty)\langle L^k/z\rangle(\mathbf{I} \mid Q_1)Q_2 \cdots Q_p \\ & \to_r & xz^\infty\langle L^k/z\rangle\langle(\mathbf{I}|Q_1)/x\rangle Q_2 \cdots Q_p \\ & \to_r & \mathbf{I}z^\infty\langle L^k/z\rangle\langle Q_1/x\rangle Q_2 \cdots Q_p \\ & \to_r & w\langle z^\infty/w\rangle\langle L^k/z\rangle\langle Q_1/x\rangle Q_2 \cdots Q_p \\ & \to_r^\star & (\lambda y.\Omega)\langle z^\infty/w\rangle\langle L^{k-1}/z\rangle\langle Q_1/x\rangle Q_2 \cdots Q_p \,. \end{array}$$

If $p = 1$, there are no more arguments to consume, and hence $C[T]\Downarrow_r$, while if $p \geqslant 2$, then

$$C[T] \;\to_r^\star\; \Omega\langle z^\infty/w\rangle\langle L^{k-1}/z\rangle\langle Q_1/x\rangle\langle Q_2/y\rangle Q_3 \cdots Q_p \Uparrow_r .$$

It should be clear that these evaluations are independent of k, provided k is at least 1. Therefore, no difference between M and N can be observed in λ_r. Instead, the λ_r^c-context $A = [\,](\lambda w.\mathrm{c}^n(w))$, where $\mathrm{c}^i(w) = \underbrace{(\mathrm{c}w)\cdots(\mathrm{c}w)}_{i \text{ times}}$ is a kind of convergence testing operator of arity i, separates M and N:

$$\begin{array}{lll} A[N] & \to_{rc}^\star & (xz^\infty)\langle L^n/z\rangle\langle \lambda w.\mathrm{c}^n(w))/x\rangle \\ & \to_{rc} & (\lambda w.\mathrm{c}^n(w))z^\infty\langle L^n/z\rangle\langle \mathbf{1}/x\rangle \\ & \to_{rc} & \mathrm{c}^n(w)\langle z^\infty/w\rangle\langle L^n/z\rangle\langle \mathbf{1}/x\rangle \\ & \to_{rc}^\star & (\mathrm{c}L)\mathrm{c}^{n-1}(w)\langle z^\infty/w\rangle\langle L^{n-1}/z\rangle\langle \mathbf{1}/x\rangle \\ & \to_{rc} & \mathbf{I}\mathrm{c}^{n-1}(w)\langle z^\infty/w\rangle\langle L^{n-1}/z\rangle\langle \mathbf{1}/x\rangle \\ & \vdots & \\ & \to_{rc} & \mathbf{I}\langle z^\infty/w\rangle\langle \mathbf{1}/z\rangle\langle \mathbf{1}/x\rangle \Downarrow_{rc} . \end{array}$$

But, since $n - m \geqslant 1$, the evaluation of $A[M]$ is deadlocked:

$$\begin{array}{lll} A[M] & \to_{rc}^\star & \mathrm{c}^n(w)\langle z^\infty/w\rangle\langle L^m/z\rangle\langle \mathbf{1}/x\rangle \\ & \to_{rc}^\star & \mathrm{c}^{n-m}(w)\langle z^\infty/w\rangle\langle \mathbf{1}/z\rangle\langle \mathbf{1}/x\rangle \Uparrow_{rc} . \end{array}$$

3. Denotational semantics

In this section, we address the following two questions: what kind of domain equation can we associate with λ-calculi with resources, and what are the solutions like? Following Scott's approach, we are led to consider domains isomorphic to (a variant of) their continuous function space (Scott 1982; Plotkin 1981). We first observe that the equation $D = (D \to D)_\perp$ for lazy λ-calculus is not suitable for resource calculi. Indeed, even if the canonical solution of this equation over complete lattices allows for an interpretation of

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

non-determinism through the binary lub, or 'join' operation (Boudol 1990)†, this operator is idempotent and hence does not allow us to distinguish infinitely available resources from finite ones. In conclusion, the standard equation is not expressive enough for enabling the interpretation of the arguments in resource calculi.

In Section 3.2 we propose the equation $D = (\mathscr{M}(D) \to D)_\bot$ for resource calculi, where D is a complete prime algebraic lattice and $\mathscr{M}(D)$ is a domain of multisets. The choice of the category (complete lattices) does not only come from the need to interpret non-determinism, but also from that of modelling the substitution process, as we will see in Section 3.3.

3.1. *Background*

We recall some standard definitions of domain theory (Amadio and Curien 1998). A *partial order* is a pair $(D, \sqsubseteq)$ where $\sqsubseteq$ is a binary relation on D that is reflexive, transitive and antisymmetric. A subset X of a partial order $(D, \sqsubseteq)$ is *directed* iff $X \neq \varnothing$ and if $d, e \in X$, then there is an $x \in X$ such that $d \sqsubseteq x$ and $e \sqsubseteq x$. A *cpo* is a partial order with least element $\bot$ such that every directed subset X has a least upper bound (l.u.b.), written $\bigsqcup X$. A *continuous* function between two cpo's is a function preserving directed l.u.b.'s. An element d of a cpo D is *compact* iff for every directed subset X of D, $d \sqsubseteq \bigsqcup X$ implies $d \sqsubseteq x$ for some member x of X. For instance, $\bot$ is compact. We will use $\mathscr{K}(D)$ to denote the set of compact elements of D. A cpo D is *algebraic* iff for every element x, the set $\{d \in D \mid d \in \mathscr{K}(D) \text{ and } d \sqsubseteq x\}$ is directed and has l.u.b. x. It is *ω-algebraic* if it is algebraic and has denumerably many compact elements.

A *complete lattice* is a partial order $(D, \sqsubseteq)$ such that each subset X of D has a least upper bound $\bigsqcup X$. The binary l.u.b.'s are written $x \sqcup y$. A complete lattice has a least and a greatest element, namely $\bot = \bigsqcup \varnothing$ and $\top = \bigsqcup D$. In this framework, an element c is compact iff for any $X \subseteq D$ we have $c \sqsubseteq \bigsqcup X \Rightarrow c \sqsubseteq \bigsqcup Y$ for some finite subset Y of X. An element p of a lattice is *prime* iff for any finite subset Y of D such that $p \sqsubseteq \bigsqcup Y$ there exists $x \in Y$ such that $p \sqsubseteq x$. Notice that $\bot$ is not prime. We will denote by $\mathscr{K}\mathscr{P}(D)$ the set of compact prime elements of D. We write $\mathscr{K}\mathscr{P}_\bot(D) = \mathscr{K}\mathscr{P}(D) \cup \{\bot\}$. A *prime algebraic lattice* (p.a.l.) (Nielsen *et al.* 1981) is a complete lattice D such that any element x of D is the join of the compact primes it dominates:

$$x = \bigsqcup \{c \mid c \in \mathscr{K}\mathscr{P}(D) \text{ and } c \sqsubseteq x\}.$$

Note that the definition of p.a.l. could have been given by replacing $\mathscr{K}\mathscr{P}(D)$ with $\mathscr{K}\mathscr{P}_\bot(D)$.

A *downset* over a partial order $(D, \sqsubseteq)$ is a subset X such that

$$x \in X \text{ and } y \sqsubseteq x \;\Rightarrow\; y \in X.$$

The *downset completion* of a partial order $(D, \sqsubseteq)$ is the set of downsets of D ordered by inclusion, which is a prime algebraic lattice. The set of non-empty downsets is called the

† The solution over the complete lattices is an adequate model of lazy λ-calculus and a fully abstract model of the lazy calculus augmented with c and ⊕.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

non-empty downset completion. A non-empty set $X \subseteq D$ is an *upperset* over $(D, \sqsubseteq)$ if and only if

$$x \in X \text{ and } x \sqsubseteq y \quad \Rightarrow \quad y \in X.$$

We denote by $\uparrow X$ the least upperset of D containing X; for singletons $\{x\}$ the notation becomes $\uparrow x$.

Proposition 3.1. Any prime algebraic lattice $(D, \sqsubseteq)$ is isomorphic to the downset completion of its set of compact prime elements $(\mathscr{K}\mathscr{P}(D), \sqsubseteq)$. Alternatively, any prime algebraic lattice $(D, \sqsubseteq)$ is isomorphic to the non-empty downset completion of $(\mathscr{K}\mathscr{P}_\perp(D), \sqsubseteq)$.

Proof. Let c and p be the transformations defined by

$$x \in D \quad \Rightarrow \quad c(x) = \{y \mid y \in \mathscr{K}\mathscr{P}(D) \text{ and } y \sqsubseteq x\}.$$

$$d \text{ downset on } \mathscr{K}\mathscr{P}(D) \quad \Rightarrow \quad p(d) = \bigsqcup d.$$

It is clear that $c(x)$ is a downset and that $p(d)$ is in D since every subset has a l.u.b. Moreover, $x = \bigsqcup\{e \mid e \in \mathscr{K}\mathscr{P}(D) \text{ and } e \sqsubseteq x\}$ implies $p(c(x)) = x$. It remains to prove that $c(p(d)) = d$. Observe that

$$\begin{array}{rcll} c(p(d)) & = & \{y \mid y \in \mathscr{K}\mathscr{P}(D) \text{ and } y \sqsubseteq \bigsqcup d\} & \\ & = & \{y \mid y \in \mathscr{K}\mathscr{P}(D) \text{ and } y \sqsubseteq x \text{ and } x \in d\} & \\ & = & \{y \mid y \in \mathscr{K}\mathscr{P}(D) \text{ and } y \in d\} & (d \text{ downset}) \\ & = & d & (d \subseteq \mathscr{K}\mathscr{P}(D)). \end{array}$$ □

An *ideal* is a directed downset. The *ideal completion* **Ideal**$(D, \sqsubseteq)$ of a partial order $(D, \sqsubseteq)$ is the set of ideals of D ordered by inclusion, which is algebraic. Moreover, any algebraic cpo D is order-isomorphic to **Ideal**$(\mathscr{K}(D), \sqsubseteq)$. A *filter* is a co-directed upperset, that is, a non-empty set $X \subseteq D$ is a filter over $(D, \sqsubseteq)$ iff it is an upperset over this poset and

$$\forall x, y \in X\ \exists z \in X\ (z \sqsubseteq x \text{ and } z \sqsubseteq y).$$

Given $a \in D$, the least filter containing a is $\uparrow a = \{x \mid a \sqsubseteq x\}$.

We will also be concerned with the function space construction and with lifted domains. Given cpos D and E, their function space $D \to E$ is the cpo of all continuous functions from D to E with the pointwise ordering

$$f \leqslant g \stackrel{def}{\Longleftrightarrow} \forall x \in D\ f(x) \sqsubseteq_E g(x).$$

We use a λ-like syntax to describe functions in $(D \to E)$: where $\lambda v \in D.e$ stands for the function f such that for any $d \in D$, we have that $f(d)$ is the element of E obtained by replacing v by d in e. The strict function space, $D \to_\perp E$, is the set $\{f : D \to E \mid f(\perp) = \perp\}$, whose elements are called strict continuous functions, with the pointwise ordering inherited from $D \to E$. The lifting $D_\perp$ of a partial order D is the set $\{\langle 0, d\rangle \mid d \in D\} \cup \{\perp\}$ with the ordering

$$x \leqslant y \text{ iff } (x = \perp) \text{ or } (\exists d, d' \in D\ d \sqsubseteq d' \text{ and } x = \langle 0, d\rangle \text{ and } y = \langle 0, d'\rangle).$$

We often simply write d for $\langle 0, d\rangle$. Two continuous functions, *up* and *down*, are associated with the lifting construction. Their type and definition are as follows:

$$up \quad : \quad D \to D_\perp \qquad\qquad down \quad : \quad D_\perp \to_\perp D$$

$$up(d) \quad = \quad \langle 0, d\rangle \qquad\qquad down(\langle 0, d\rangle) \quad = \quad d\ ,\ down(\perp) = \perp_D\ .$$

The *lifted application* $\_(\!(\_)\!) : (D \to E)_\perp \times D \to E$ is the uncurried form of *down*:

$$f(\!(d)\!) = down(f)(d) \qquad \text{with } f \in (D \to E)_\perp \text{ and } d \in D.$$

$\_(\!(\_)\!) : (D \to E)_\perp \times D \to E$ is the uncurried form of *down* :

$$f(\!(d)\!) = down(f)(d) \qquad \text{with } f \in (D \to E)_\perp \text{ and } d \in D.$$

3.2. *A domain equation for resource calculi*

The aim of this section is to give a precise definition of the aforementioned domain equation $D = (\mathscr{M}(D) \to D)_\perp$. We base our construction of $\mathscr{M}(D)$ on the observation that it should represent multisets in the same way as powerdomains allow us to represent sets.

Given a set E, let $\mathscr{M}_f(E)$ denote the set of finite multisets of elements of E. The multiset union of u and v is denoted $u \centerdot v$ and we identify an element $e \in E$ with the singleton multiset $\{e\}$. Then any multiset is a product $u = e_1 \centerdot \cdots \centerdot e_n$ with $n \geqslant 0$ and possibly with repetitions. The empty product is the empty multiset $\varnothing$, and the product operation $\centerdot$ is commutative and associative. We may also write multisets as monomials $u = e_1^{m_1} \centerdot \cdots \centerdot e_k^{m_k}$, where the m_i's are positive integers. Let $(P, \sqsubseteq)$ be a poset; we define the poset of finite multisets $(\mathscr{M}_f(P), \lessdot)$ with least element $\varnothing$ as follows. The preorder $\lessdot$ is the least precongruence containing $\sqsubseteq$ and the inclusion of multisets. That is, the least preorder on $\mathscr{M}_f(P)$ satisfying

$$\begin{array}{rcll} p & \lessdot & q & \text{if } p \sqsubseteq q \\ u & \lessdot & u \centerdot v & \\ u \centerdot v & \lessdot & u' \centerdot v' & \text{if } u \lessdot u' \text{ and } v \lessdot v'\ . \end{array}$$

We assume that $q_1 \centerdot \cdots \centerdot q_n$ denotes the empty multiset when $n = 0$. This convention is similar to that already adopted for bags. It is easy to check that $p_1 \centerdot \cdots \centerdot p_k \lessdot u$ if and only if

$$\exists q_1, \ldots, q_k\ \forall i\ (p_i \sqsubseteq q_i \ \text{ and }\ (u = q_1 \centerdot \cdots \centerdot q_k \ \text{ or }\ \exists v\ u = q_1 \centerdot \cdots \centerdot q_k \centerdot v))\ .$$

Lemma 3.2. Let $(P, \sqsubseteq)$ be a poset. The relation $\lessdot$ is a partial order over $\mathscr{M}_f(P)$.

Proof. Let $|u|$ be the cardinality of multiset u. Then $u \lessdot v$ implies $|u| \leqslant |v|$. We show $(u \lessdot v \lessdot u \Rightarrow u = v)$ by induction on $|u|$. If $|u| = 1$, we have $|v| = 1$, so $(u \sqsubseteq v \sqsubseteq u \Rightarrow u = v)$. If $|u| > 1$, let p be a maximal element of u with respect to $\sqsubseteq$ and $u = p \centerdot u'$. There exist q and v' such that $v = q \centerdot v'$, where $p \sqsubseteq q$ and $u' \lessdot v'$. There also exist p', u'' such that $u = p' \centerdot u''$ with $q \sqsubseteq p'$ and $v' \lessdot u''$. From $p \sqsubseteq q \sqsubseteq p'$, we get $p = p' = q$ because p is maximal. Hence $u' = u''$, and then $u' = v'$, by the induction hypothesis. □

It should be clear that $\mathscr{M}(D)$ will somehow involve the operation $\mathscr{M}_f(\,)$. Instead of

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

taking $(\mathscr{K}(D) - \bot, \sqsubseteq)$ as is classical in the construction of powerdomains, we assume that D is a prime algebraic lattice, and then we define

$$\mathscr{M}(D) = \mathbf{Ideal(Bag}(D), \lessdot\,\mathbf{)} \quad \text{where} \quad \mathbf{Bag}(D) = \mathscr{M}_f(\mathscr{K}\mathscr{P}(D)).$$

In this construction, $(\mathbf{Bag}(D), \lessdot\,)$ is the analog of the finite powerset ordered by the lower order used to build up lower powerdomains.

An alternative presentation of $(\mathbf{Bag}(D), \lessdot\,)$ in terms of non-empty multisets comes up naturally by associating the empty multiset to the bottom $\bot$ of D. Given a poset $(P, \sqsubseteq)$ with bottom element $\bot$, let $\mathscr{M}^*_f(P)$ be the set of finite non-empty multisets of P and let $\lessdot_\bot$ be the preorder defined as $\lessdot$, plus $\bot$ as neutral element, that is, $\bot \centerdot u \lessdot_\bot u$. Notice that in this setting, $\bot$ is the least non-empty multiset: $\bot \lessdot_\bot u$ for any non-empty multiset u just because $\bot \sqsubseteq p$ for any p in P.

Proposition 3.3. The posets $(\mathscr{M}_f(P), \lessdot\,)$ and $(\mathscr{M}^*_f(P_\bot), \lessdot_\bot\,)$ are isomorphic.

Proof. It is easy to see that the two injections b, b' defined by

$$(\mathscr{M}_f(P), \lessdot\,) \xrightarrow{b} (\mathscr{M}^*_f(P_\bot), \lessdot_\bot\,) \quad (\mathscr{M}^*_f(P_\bot), \lessdot_\bot\,) \xrightarrow{b'} (\mathscr{M}_f(P), \lessdot\,)$$

$$b(\varnothing) = \bot \qquad\qquad b'(\bot) = \varnothing \quad b'(\bot \centerdot u) = b'(u)$$

$$b(p) = p \quad b(p \centerdot u) = p \centerdot b(u) \qquad b'(p) = p \quad b'(p \centerdot u) = \begin{cases} p \centerdot b'(u) & \text{if } b'(u) \neq \varnothing \\ p & \text{otherwise} \end{cases}$$

where $p \in P$ and $u \in \mathscr{M}_f(P)$ or $u \in \mathscr{M}^*_f(P)$, verify the following two statements:

$$u, v \in \mathscr{M}_f(P) \text{ and } u \lessdot v \;\Rightarrow\; b(u) \lessdot_\bot b(v) \text{ and}$$

$$u, v \in \mathscr{M}^*_f(P_\bot) \text{ and } u \lessdot_\bot v \;\Rightarrow\; b'(u) \lessdot b'(v)\,. \qquad \square$$

In view of this isomorphism, the domain equation $D = (\mathscr{M}(D) \to D)_\bot$ becomes

$$D = (\mathbf{Ideal(}\mathscr{M}^*_f(\mathscr{K}\mathscr{P}_\bot(D)), \lessdot_\bot\,\mathbf{)} \to D)_\bot\,.$$

Our choice of compact primes instead of compact elements to carry out the construction $\mathscr{M}(D)$ allows us to define a mapping j transforming an ideal of $\mathscr{M}(D)$ into an element of D for the interpretation of variables (see Section 3.3). Recall that any p.a.l. D is isomorphic to the non-empty down-set completion of $\mathscr{K}\mathscr{P}_\bot(D)$. Given an ideal of multisets, j forms the union of their underlying sets. Formally, if $a \in \mathscr{M}(D)$:

$$j(a) = \bigsqcup \{ x \in \mathscr{K}\mathscr{P}_\bot(D) \mid \exists u \in a \;\; x \lessdot u \}\,.$$

The mapping j on $\mathscr{K}(\mathscr{M}(D))$ has the following shape: let a be the principal ideal $\downarrow(d_1 \centerdot \cdots \centerdot d_n)$ with $d_i \in \mathscr{K}\mathscr{P}_\bot(D)$, then

$$j(a) = d_1 \sqcup \cdots \sqcup d_n\,.$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

3.3. *Interpretation*

We explain in which sense the solutions of the domain equation defined in the previous section constitute models of λ-calculi with resources.

The interpretation function $\mathscr{V}[\![M]\!]_\rho$ for λ_r^c takes as arguments a term M and an environment ρ closing M, and gives back an element of D. Environments are mappings in Env $= [Var \to \mathscr{M}(D)]$ with a finite domain. We call CEnv the collection of compact environments, that is, those environments defined on $[Var \to \mathscr{K}(\mathscr{M}(D))]$.

Since $\mathscr{M}(D)$ does not have arbitrary l.u.b.'s, it does not allow for a direct interpretation of arguments. The meaning of a term relies on the meaning of the *finite terms* related to it. By finite terms we mean terms whose bags have a finite number of components. We define a binary relation $P \propto Q$ that captures the idea that every multiplicity in bag P becomes finite in bag Q – all infinite multiplicities of P are replaced in Q by finite ones, and finite multiplicities in P may decrease their values in Q:

$$M^n \propto \begin{cases} \mathbf{1} & \\ M^m & \text{for all } m \in \mathbb{N} \text{ if } n = \infty \\ M^m & \text{for all } m \leqslant n \text{ if } n \in \mathbb{N} \end{cases}$$

$$P \propto P' \text{ and } Q \propto Q' \;\Rightarrow\; (P \mid Q) \propto (P' \mid Q')\,.$$

Some additional notation is needed to define the interpretation function. For any finite collection V of variables, the environment $\rho/_V$ is such that

$$\rho/_V(y) = \begin{cases} \rho(y) & \text{if } y \notin V \\ \text{undefined} & \text{otherwise.} \end{cases}$$

In order to simplify the notation, let us define the *product of compact environments*, $\rho_m \centerdot \cdots \centerdot \rho_n$, with $m \leqslant n$, by

$$(\forall i\ \rho_i \in \text{CEnv} \;\Rightarrow\; (\forall x\ \rho_i(x) = \downarrow u_i \;\Rightarrow\; (\rho_m \centerdot \cdots \centerdot \rho_n)(x) = \downarrow(u_m \centerdot \cdots \centerdot u_n)))\,.$$

The definition of the interpretation of terms with resources is given in Figure 5.

The usual interpretation $\mathscr{V}[\![x]\!]_\rho = \rho(x)$ does not work, because values for variables are taken from $\mathscr{M}(D)$ and not from D. Whence the use of the mapping $j : \mathscr{M}(D) \to D$ in the equation $\mathscr{V}[\![x]\!]_\rho = j(\rho(x))$. The interpretation of the abstraction is as usual in a lazy framework. The remaining constructs are application and explicit substitution, for which the environment (that is, the bags of terms associated to the free variables) should not be duplicated. Instead, environments are split and distributed to free occurrences of variables in all possible ways. Moreover, it should be obvious that any convergent evaluation of a term uses the arguments occurring in it only a finite number of times each; hence it seems natural to compute its denotation in terms of its associated finite terms. All these contributions are collected together by means of a l.u.b. which due to the non-determinacy of evaluation is not directed. This is why we require that D is a complete lattice. Notice that the environments $\rho_0, \ldots, \rho_n$ used to give the semantics of application and explicit substitution are compact ones (see the definition of the product $\rho_m \centerdot \cdots \centerdot \rho_n$). Certainly, we could have used arbitrary environments, but our choice eases technical proofs.

4. The model of uppersets in logical form

The aim of this section is to construct a concrete logical model of resource calculi in the style of logical filter models for pure (lazy) λ-calculus (see Barendregt *et al.* (1983), Coppo *et al.* (1984) and Ronchi della Rocca (1993), and also Boudol (1990), Boudol (1991), Dezani *et al.* (1994), Dezani *et al.* (1999a) and Dezani *et al.* (1999b)), which turns out to be a presentation of the canonical, or minimal, or initial, solution of the domain equation defined in Section 3.2 (Pitts 1996; Amadio and Curien 1998). We shall not prove, nor even define more precisely, canonicity here; the details would closely follow the treatment given in, say, Abramsky and Ong (1993). The underlying logic consists of a language of types together with an entailment relation between types. The meaning of a term is a set of formulas of the logic.

4.1. *Type theory*

The syntax for types (formulas) is determined by the following grammar:

$$\begin{array}{llcl} (\mathrm{Ft}) & \phi & ::= & \omega \mid \pi \to \phi \\ (\mathrm{Fb}) & \pi & ::= & \phi \mid (\pi \times \pi)\,. \end{array}$$

Unless stated otherwise, ϕ, σ, δ will range over Ft, and π, ψ, θ will range over Fb.

The type ω stands for the truth in the logic; it is the least piece of information we can

$$\mathscr{V}[\![x]\!]_\rho = j(\rho(x))$$

$$\mathscr{V}[\![\lambda x.M]\!]_\rho = up(\lambda u \in \mathscr{M}(D).\mathscr{V}[\![M]\!]_{\rho[x:=u]})$$

$$\mathscr{V}[\![MP]\!]_\rho = \bigsqcup \mathscr{V}[\![M]\!]_{\rho_0}(\!(\downarrow d_1 \centerdot \cdots \centerdot d_n)\!) \quad \text{where} \begin{cases} \rho \supseteq \rho_0 \centerdot \rho_1 \centerdot \cdots \centerdot \rho_n, \\ P \propto (M_1 \mid \cdots \mid M_n) \text{ and} \\ d_i \leqslant \mathscr{V}[\![M_i]\!]_{\rho_i} \cap \mathscr{K}\mathscr{P}_\bot(D) \end{cases}$$

$$\mathscr{V}[\![M\langle P/x\rangle]\!]_\rho = \bigsqcup \mathscr{V}[\![M]\!]_{\rho_0[x:=\downarrow d_1 \centerdot \cdots \centerdot d_n]} \quad \text{where} \begin{cases} \rho \supseteq \rho_0/_x \centerdot \rho_1 \centerdot \cdots \centerdot \rho_n, \\ P \propto (M_1 \mid \cdots \mid M_n) \text{ and} \\ d_i \leqslant \mathscr{V}[\![M_i]\!]_{\rho_i} \cap \mathscr{K}\mathscr{P}_\bot(D) \end{cases}$$

$$\mathscr{V}[\![\mathrm{c}P]\!]_\rho = \begin{cases} \mathscr{V}[\![\mathbf{I}]\!]_\rho & \text{if } P \equiv (M \mid Q) \text{ and } \mathscr{V}[\![M]\!]_\rho \neq \bot \\ \bot & \text{otherwise} \end{cases}$$

Fig. 5. Interpretation function

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

have about a term. Arrow types are in a sense implicative formulas used to give meaning to functions. The language for argument types involves a novel constructor $\times$, which is the logical counterpart of parallel composition. This 'product' of types acts as a kind of conjunction, allowing us to group type information about the arguments of a function. For instance $((\pi \to \phi) \times \pi) \to \phi$ will be among the types of $\lambda x.(xx)$. However, unlike conjunction, product is not idempotent, that is, $\phi \times \phi$ is not equivalent to ϕ in general (this is the case for $\phi = \omega$ only). If both ϕ_0 and ϕ_1 are information about a term M, we cannot say that $\phi_0 \times \phi_1$ is; what we can say is that this product type is part of the meaning of the finite bag $(M \mid M)$.

The entailment relation $\phi \leqslant \sigma$, which can be read as 'ϕ implies σ', is the least reflexive and transitive relation on Fb verifying the usual laws for arrows in the weak theory of λ-calculus (1 to 3), together with additional laws for product (4 to 8):

1. $\phi \leqslant \omega$
2. $\pi \to \omega \leqslant \omega \to \omega$
3. $\pi_1 \leqslant \pi_0$ and $\phi_0 \leqslant \phi_1 \;\Rightarrow\; (\pi_0 \to \phi_0) \leqslant (\pi_1 \to \phi_1)$

4. $\omega \times \pi \leqslant \pi$
5. $\pi \leqslant \omega \times \pi$
6. $\pi_0 \times \pi_1 \leqslant \pi_1 \times \pi_0$
7. $\pi_0 \times (\pi_1 \times \pi_2) \leqslant (\pi_0 \times \pi_1) \times \pi_2$
8. $\pi_0 \leqslant \pi_0'$ and $\pi_1 \leqslant \pi_1' \;\Rightarrow\; \pi_0 \times \pi_1 \leqslant \pi_0' \times \pi_1'$.

Note that $\times$ does not verify idempotency, that is, we do not assume

$$\pi \leqslant \pi_0 \text{ and } \pi \leqslant \pi_1 \Rightarrow \pi \leqslant (\pi_0 \times \pi_1).$$

However, $\times$ verifies the other usual properties of conjunction:

$$\pi_0 \times \pi_1 \leqslant \pi_0 \quad \text{and} \quad \pi_0 \times \pi_1 \leqslant \pi_1.$$

We write $\phi \sim \sigma$ whenever $\phi \leqslant \sigma$ and $\sigma \leqslant \phi$ are provable. For instance

$$\pi \to \omega \sim \omega \to \omega\,,\ \omega \times \pi \sim \pi\,,\ \pi_0 \times \pi_1 \sim \pi_1 \times \pi_0\,,\ \text{and } (\pi_0 \times \pi_1) \times \pi_2 \sim \pi_0 \times (\pi_1 \times \pi_2).$$

The associativity of $\times$ allows us to discard parentheses from product types, whose general shape is $\sigma_1 \times \cdots \times \sigma_n$ with $n \geqslant 1$ and $\sigma_i \in$ Ft. Finally, observe that for any $\sigma \in$ Ft we have

$$(\omega \leqslant \sigma \;\Rightarrow\; \sigma = \omega) \;\text{ and }\; (\sigma \neq \omega \;\Rightarrow\; \sigma \leqslant \omega \to \omega)\,.$$

The type theory verifies what we call the product and the arrow properties, which characterize formulas related by $\leqslant$.

Definition 4.1. We define the relation $\pi \triangleright \phi_1, \ldots, \phi_m$, which expresses the fact that $\phi_1, \ldots, \phi_m$ are the meaningful factors (that is, different from ω) of π, as follows:

- $\omega \triangleright \varepsilon$ (the empty sequence)
- $\psi \to \delta \triangleright \psi \to \delta$
- $\pi \triangleright \phi_1, \ldots, \phi_m$ and $\pi' \triangleright \phi_1', \ldots, \phi_k' \;\Rightarrow\; \pi \times \pi' \triangleright \phi_1, \ldots, \phi_m, \phi_1', \ldots, \phi_k'$.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Proposition 4.2. (Product property) Let $\pi \rhd \phi_1, \ldots, \phi_n$ and $\psi \rhd \sigma_1, \ldots, \sigma_m$. The following equivalence holds:

$$\pi \leqslant \psi \iff \exists i : [1, m] \to [1, n] \ (i \text{ injective and } (\forall j \in [1, m] \ \phi_{i(j)} \leqslant \sigma_j)) .$$

Moreover, the size of the proof of $\phi_{i(j)} \leqslant \sigma_j$ is at most equal to that of $\pi \leqslant \psi$.

Proof. ($\Leftarrow$) If $m = 0$, then $\psi = \omega \times \cdots \times \omega$, hence $\pi \leqslant \psi$ holds for any π by Clauses 1, 4 and 8 of the definition of $\leqslant$. Otherwise,

$$\phi_{i(1)} \times \cdots \times \phi_{i(m)} \leqslant \sigma_1 \times \cdots \times \sigma_m \sim \psi \quad \text{by 8.}$$

Moreover, $\pi \leqslant \phi_{i(1)} \times \cdots \times \phi_{i(m)}$ by commutativity, 4 and 8. Hence $\pi \leqslant \psi$ by transitivity.

($\Rightarrow$) We proceed by structural induction on the size l of the derivation of $\pi \leqslant \psi$:

$l = 1$: If $\pi \leqslant \psi$ is an instance of reflexivity or Axioms 1 or 2, or 4 to 7, then $n = m$ and the sequences $\phi_1, \ldots, \phi_m$ and $\sigma_1, \ldots, \sigma_m$ are equal up to permutations. That is there exists a bijection $i : [1, m] \to [1, m]$ such that $\phi_{i(j)} = \sigma_j$ for all $j = 1, \ldots, m$.

$l > 1$: Three cases arise depending on whether the last rule used in the derivation of $\pi \leqslant \psi$ is 3, in which case there is nothing to prove, or 8 or transitivity. In the second case, $\pi = \pi_0 \times \pi_1$ and $\psi = \psi_0 \times \psi_1$ with $\pi_i \leqslant \psi_i$ in l_i steps, for $i = 0, 1$. We have $l = l_0 + l_1$. Let

$$\begin{array}{ll} \pi_0 \rhd \phi_1, \ldots, \phi_a & \pi_1 \rhd \phi_{a+1}, \ldots, \phi_n \\ \psi_0 \rhd \sigma_1, \ldots, \sigma_b & \psi_1 \rhd \sigma_{b+1}, \ldots, \sigma_m . \end{array}$$

By the induction hypothesis, there exist injections $k_0 : [1, b] \to [1, a]$ and $k_1 : [b+1, m] \to [a+1, n]$ such that $\phi_{k_h(j)} \leqslant \sigma_j$ in at most l_h steps for any j, with $h = 0$ or $h = 1$. Define $i : [1, m] \to [1, n]$ by

$$i(j) = \begin{cases} k_0(j) & \text{if } 1 \leqslant j \leqslant b \\ k_1(j) & \text{if } b + 1 \leqslant j \leqslant m. \end{cases}$$

Then $\phi_{i(j)} \leqslant \sigma_j$ in less than l steps. Finally, if $\pi \leqslant \psi$ is proved by transitivity,

$$\pi \leqslant \theta \text{ in } l_1 \text{ steps and } \theta \leqslant \psi \text{ in } l_2 \text{ steps,}$$

with $l_1 + l_2 + 1 = l$. Let $\theta \rhd \delta_1, \ldots, \delta_p$. By induction, there exist two injections $k_1 : [1, p] \to [1, n]$ and $k_2 : [1, m] \to [1, p]$ such that $\phi_{k_1(j)} \leqslant \delta_j$ for all $j = 1, \ldots, p$ and $\delta_{k_2(j)} \leqslant \sigma_j$ for all $j = 1, \ldots, m$. Define $i = k_1 \circ k_2 : [1, m] \to [1, n]$. From the induction hypotheses,

$$\begin{array}{rcll} \phi_{k_1(k_2(j))} & \leqslant & \delta_{k_2(j)} & \text{in at most } l_1 \text{ steps and} \\ \delta_{k_2(j)} & \leqslant & \sigma_j & \text{in at most } l_2 \text{ steps.} \end{array}$$

By transitivity, $\phi_{k_1(k_2(j))} \leqslant \sigma_j$ in at most l steps. □

Proposition 4.3. (Arrow property) For all π, ψ, ϕ, σ,

$$(\pi \to \phi \leqslant \psi \to \sigma) \iff (\sigma \neq \omega \Rightarrow \psi \leqslant \pi \text{ and } \phi \leqslant \sigma) .$$

Proof. ($\Leftarrow$) By cases on σ. If $\sigma = \omega$, we have

$$\pi \to \phi \leqslant \pi \to \omega \leqslant \omega \to \omega \leqslant \psi \to \omega$$

for any π, ϕ, ψ. If $\sigma \neq \omega$, the statement follows immediately by Clause 3.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

($\Rightarrow$) Assume $\sigma \neq \omega$, and proceed by induction on the size l of the derivation of $\pi \to \phi \leqslant \psi \to \sigma$. If $l = 1$, then $\pi \to \phi \leqslant \psi \to \sigma$ is established by reflexivity, hence $\pi = \psi$ and $\phi = \sigma$. Otherwise (that is, $l > 1$), the derivation is either by application of Clause 3 or by transitivity. The first case supposes $\psi \leqslant \pi$ and $\phi \leqslant \sigma$ directly. In the second case, there exists θ such that

$$\pi \to \phi \leqslant \theta \quad \text{and} \quad \theta \leqslant \psi \to \sigma,$$

with sizes l_1 and l_2, respectively. The product property implies $\theta \rhd \pi' \to \phi'$ (θ has only one factor different from ω). That is, $\theta \sim \pi' \to \phi'$. Thus we have $\pi \to \phi \leqslant \pi' \to \phi' \leqslant \psi \to \sigma$. By induction, $\psi \leqslant \pi'$ and $\phi' \leqslant \sigma$ hold. Moreover, $\phi' \neq \omega$, since $\omega \leqslant \sigma$ would imply $\sigma = \omega$, contradicting the hypothesis. Then by induction again, we have $\phi \leqslant \phi'$ and $\pi' \leqslant \pi$. Therefore, $\psi \leqslant \pi$ and $\phi \leqslant \sigma$. □

4.2. *Domain of uppersets*

In order to solve the domain equation for the λ-calculi with resources, the model is required to be a p.a.l., that is the non-empty downset completion of some poset with bottom (*cf.* Proposition 3.1). To this end, we will use the poset (Ft, $\leqslant$), which is in reverse order with respect to the standard convention: $\phi \leqslant \tau$ means that τ is less precise than ϕ. Therefore, we use the notion dual to that of non-empty downset: that of non-empty upperset, called simply upperset in the rest of the paper.

Definition 4.4. (Domain $\mathscr{U}$) $\mathscr{U}$ is the collection of uppersets over (Ft, $\leqslant$), ordered by set inclusion.

Introducing the ingredient of non-emptiness in the definition of upperset is in agreement with the intended meaning for the bottom element of the domain $\mathscr{U}$, whose canonical representative must be ω. By construction, $\langle \mathscr{U}, \subseteq \rangle$ is a prime algebraic lattice. Its least element $\perp$ is $\uparrow \omega$, which is actually $\{\omega\}$, its top element $\top$ is Ft, and $x \sqcup y = x \cup y$ and $x \sqcap y = x \cap y$. Given $A \subseteq$ Ft, we have $\uparrow A = \bigsqcup\{\uparrow a \mid a \in A\} = \bigcup\{\uparrow a \mid a \in A\}$. The compact prime elements of $\mathscr{U}$ are its principal uppersets $\uparrow \phi$ and its compact elements are finite unions of compact primes.

Remark 4.5. For any $\phi, \sigma \in$ Ft, we have $\phi \leqslant \sigma$ iff $\uparrow \sigma \subseteq \uparrow \phi$.

As a corollary, $\forall \pi \;\; \uparrow(\omega \to \omega) = \uparrow(\pi \to \omega)$: the left-to-right inclusion holds by definition (Rule 2); the right-to-left one holds because $\pi \leqslant \omega$ for any π. Furthermore, if $\phi \not\sim \omega$, then $\uparrow \phi$ contains all functions types of the shape $\pi \to \omega$, since $\phi \leqslant \omega \to \omega$.

We argue that the domain $\mathscr{U}$ is a suitable candidate to solve, up to isomorphism, the semantic equation for λ_r^c, and that, in fact, it provides an adequate model for the resource calculi. The last point will be studied in the next sections. Here we treat the first point, that is, we show the following isomorphism:

$$\mathscr{U} \sim (\mathscr{M}(\mathscr{U}) \to \mathscr{U})_\perp.$$

The proof is based on the observation that the domain $\mathscr{M}(\mathscr{U})$ has a concrete (logical) presentation over the poset (Fb, $\leqslant$).

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Definition 4.6. (Domain $\mathscr{U}_{\mathscr{B}}$) $\mathscr{U}_{\mathscr{B}}$ is the collection of filters over $(\mathrm{Fb}, \leqslant)$, ordered by set inclusion.

Remark 4.7. For any $\pi, \psi \in \mathrm{Fb}$, we have $\pi \leqslant \psi$ iff $\uparrow \psi \subseteq \uparrow \pi$.

The poset $(\mathscr{U}_{\mathscr{B}}, \subseteq)$ is an ω-algebraic cpo where the l.u.b.'s. of directed subsets of $\mathscr{U}_{\mathscr{B}}$ are their union, and whose compact filters are the principal filters $\uparrow \pi$. The next proposition states that this poset gives a concrete presentation of the domain $\mathscr{M}(\mathscr{U})$.

Proposition 4.8. $\mathscr{U}_{\mathscr{B}}$ and $\mathscr{M}(\mathscr{U})$ are order-isomorphic.

Proof. Recall that $\mathscr{M}(\mathscr{U}) = \mathbf{Ideal}(\mathscr{M}^*_f(\mathscr{K}\mathscr{P}_\perp(\mathscr{U})), \lessdot_\perp)$. It is enough to show that the sets of compact elements of $\mathscr{U}_{\mathscr{B}}$ and $\mathscr{M}(\mathscr{U})$ are order-isomorphic. These sets are the following:

$$\mathscr{K}(\mathscr{U}_{\mathscr{B}}) = \{\uparrow \pi \mid \pi \in \mathrm{Fb}\}$$

$$\mathscr{K}(\mathscr{M}(\mathscr{U})) = \{\downarrow(d_1 \centerdot \cdots \centerdot d_n) \mid n \geqslant 1 \text{ and } d_i \in \mathscr{K}\mathscr{P}_\perp(\mathscr{U})\},$$

where $\mathscr{K}\mathscr{P}_\perp(\mathscr{U}) = \{\uparrow \phi \mid \phi \in \mathrm{Ft}\}$. Define $\kappa : \mathrm{Fb} \to \mathscr{M}^*_f(\mathscr{K}\mathscr{P}_\perp(\mathscr{U}))$ as follows:

$$\begin{array}{rcll} \kappa(\phi) & = & \uparrow \phi & \text{if } \phi \in \mathrm{Ft} \\ \kappa(\pi \times \psi) & = & \kappa(\pi) \centerdot \kappa(\psi) \,. & \end{array}$$

It is easy to check that $\uparrow \pi \in \mathscr{K}(\mathscr{U}_{\mathscr{B}}) \iff \downarrow\kappa(\pi) \in \mathscr{K}(\mathscr{M}(\mathscr{U}))$. It remains to show that, for any $\pi, \psi \in \mathrm{Fb}$, we have

$$\pi \leqslant \psi \iff \kappa(\psi) \lessdot_\perp \kappa(\pi) \,.$$

($\Rightarrow$) If $\psi \sim \omega$, then $\kappa(\psi) = (\uparrow \omega)^m \lessdot_\perp \uparrow \omega \lessdot_\perp \kappa(\pi)$ for any π. Otherwise, assume $\pi \triangleright \phi_1, \ldots, \phi_n$ and $\psi \triangleright \sigma_1, \ldots, \sigma_m$. By the product property, there is an injection $i : [1, m] \to [1, n]$ such that $\phi_{i(j)} \leqslant \sigma_j$. That is, $\uparrow \sigma_j \lessdot_\perp \uparrow \phi_{i(j)}$. Hence

$$\kappa(\psi) \lessdot \uparrow \sigma_1 \centerdot \cdots \centerdot \uparrow \sigma_m \lessdot_\perp \uparrow \phi_{i(1)} \centerdot \cdots \centerdot \uparrow \phi_{i(m)} \lessdot_\perp \kappa(\pi) \,.$$

($\Leftarrow$) We use structural induction on the derivation of $\kappa(\psi) \lessdot_\perp \kappa(\pi)$. Observe that whenever $\kappa(\theta) = u \centerdot v$, there exist θ_0, θ_1 such that $\theta = \theta_0 \times \theta_1$, $\kappa(\theta_0) = u$ and $\kappa(\theta_1) = v$.

- If $\kappa(\psi) \lessdot_\perp \kappa(\pi)$ is proved by reflexivity, then ψ and π are equal up to commutativity, and hence $\pi \leqslant \psi$.
- If both π and ψ are in Ft, then $\kappa(\psi) \lessdot_\perp \kappa(\pi)$ boils down to $\uparrow \psi \subseteq \uparrow \pi$, and we have $\pi \leqslant \psi$.
- If $\kappa(\psi) = \kappa(\pi) \centerdot \perp \lessdot_\perp \kappa(\pi)$, then $\psi = \pi \times \omega$. So $\pi \leqslant \psi$ holds.
- If $\kappa(\psi) \lessdot_\perp \kappa(\psi) \centerdot \kappa(\pi') = \kappa(\pi)$, then $\pi = \psi \times \pi' \leqslant \psi \times \omega \leqslant \psi$ holds.
- If $\kappa(\psi) = \kappa(\psi_0) \centerdot \kappa(\psi_1) \lessdot_\perp \kappa(\pi_0) \centerdot \kappa(\pi_1) = \kappa(\pi)$ comes from $\kappa(\psi_0) \lessdot_\perp \kappa(\pi_0)$ and $\kappa(\psi_1) \lessdot_\perp \kappa(\pi_1)$, then we have $\pi_0 \leqslant \psi_0$ and $\pi_1 \leqslant \psi_1$ by induction. So $\pi = \pi_0 \times \pi_1 \leqslant \psi_0 \times \psi_1 = \psi$ holds.
- If $\kappa(\psi) \lessdot_\perp \kappa(\theta) \lessdot_\perp \kappa(\pi)$, the induction hypothesis gives $\theta \leqslant \psi$ and $\pi \leqslant \theta$. Hence $\pi \leqslant \psi$ by transitivity of $\leqslant$. □

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

In order to prove that $\mathcal{U}$ and $(\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_\perp$ are isomorphic, we need to represent the continuous functions from $\mathcal{U}_{\mathcal{B}}$ to $\mathcal{U}$ by elements of $\mathcal{U}$. To this end, we successively define *step functions* (used to determine the compact elements of the domain $\mathcal{U}_{\mathcal{B}} \to \mathcal{U}$), the application operator of the model, and functions F and G that realize the isomorphism.

Definition 4.9. Step functions $f_{\pi\phi}$ from $\mathcal{U}_{\mathcal{B}}$ to $\mathcal{U}$ are given by

$$f_{\pi\phi}(c) = \begin{cases} \uparrow\phi & \text{if} \uparrow\pi \subseteq c \\ \uparrow\omega & \text{otherwise.} \end{cases}$$

Proposition 4.10. The compact elements of $(\mathcal{U}_{\mathcal{B}} \to \mathcal{U})$ are exactly given by finite least upper bounds of the shape $\bigsqcup_I f_{\pi_i\phi_i}$.

Proof. Let $f \in (\mathcal{U}_{\mathcal{B}} \to \mathcal{U})$. We show the following two properties:

1 $f_{\pi\phi} \subseteq f \iff \uparrow\phi \subseteq f(\uparrow\pi)$, and
2 $f = \bigcup\{f_{\pi\phi} \mid f_{\pi\phi} \subseteq f\}$.

The $(\Rightarrow)$ direction of (1) is easy: $f_{\pi\phi} \subseteq f \Rightarrow f_{\pi\phi}(\uparrow\pi) \subseteq f(\uparrow\pi) \iff \uparrow\phi \subseteq f(\uparrow\pi)$. As for $(\Leftarrow)$, let $\uparrow\phi \subseteq f(\uparrow\pi)$. It is enough to consider the case of $\uparrow\pi \subseteq c$, since $\uparrow\omega$ is always included in $f(c)$. By the monotonicity of f, we have $f_{\pi\phi}(c) = \uparrow\phi \subseteq f(\uparrow\pi) \subseteq f(c)$.

Note that the domain $(\mathcal{U}_{\mathcal{B}} \to \mathcal{U})$ is a p.a.l. where the l.u.b's are defined pointwise, that is $(\bigcup g)(x) = \bigcup g(x)$. Then, using (1), proving (2) is the same as proving the following equation for any $x \in \mathcal{U}_{\mathcal{B}}$:

$$f(x) = \bigcup\{f_{\pi\phi}(x) \mid \uparrow\phi \subseteq f(\uparrow\pi)\}.$$

The right-hand side of the equation can be read as

$$A = \bigcup\{\uparrow\phi \mid \exists\pi \ \uparrow\phi \subseteq f(\uparrow\pi) \text{ and } \uparrow\pi \subseteq x\}.$$

Exploiting the fact that f is continuous and that any x is equal to the l.u.b. of the directed set $\{\uparrow\pi \mid \uparrow\pi \subseteq x\}$, we can check that $A = \bigcup\{\uparrow\phi \mid \uparrow\phi \subseteq f(x)\}$, which is $f(x)$ since $\mathcal{U}$ is a p.a.l.

Clearly, finite l.u.b.'s of step functions are compact, and it follows immediately from (2) that all compact functions have this form. □

Definition 4.11. (Application in the model) The *application* $\_\cdot\_ : \mathcal{U} \times \mathcal{U}_{\mathcal{B}} \to \mathcal{U}$ is the continuous function defined by

$$x \cdot y = \begin{cases} \{\sigma \in \mathrm{Ft} \mid \exists\pi \in y\,.\,(\pi \to \sigma) \in x\} & \text{if } x \neq \uparrow\omega \\ \{\omega\} & \text{otherwise.} \end{cases}$$

Corollary 4.12. If $x \neq \uparrow\omega$, then $\phi \in x \cdot \uparrow\pi \iff \pi \to \phi \in x$.

Proof. $(\Rightarrow)$ By definition of application, there is $\psi \in \uparrow\pi$ such that $\psi \to \phi \in x$, and hence $\pi \leqslant \psi$ implies $\psi \to \phi \leqslant \pi \to \phi \in x$, since x is upward closed. The $(\Leftarrow)$ direction is even simpler. □

Definition 4.13. Let F and G be the functions defined by

$$\mathcal{U} \xrightarrow{F} (\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_\perp \xrightarrow{G} \mathcal{U}$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

$$F(x) = \begin{cases} up(\lambda y \in \mathcal{U}_{\mathcal{B}}.x \cdot y) & \text{if } x \neq \uparrow \omega \\ \perp & \text{otherwise} \end{cases}$$

$$G(f) = \begin{cases} \uparrow \{\pi \to \phi \mid \phi \in f(\!(\uparrow \pi)\!)\} & \text{if } f \neq \perp \\ \uparrow \omega & \text{otherwise.} \end{cases}$$

As an immediate consequence of the definition, we have

$$F(x)(\!(d)\!) = \left\{ \begin{array}{ll} x \cdot d & \text{if } x \neq \uparrow \omega \\ \uparrow \omega & \text{otherwise} \end{array} \right\} = x \cdot d\,.$$

We also note that application is continuous as a function of its second argument and preserves arbitrary l.u.b.'s as a function of its first argument, and that G preserves arbirtrary l.u.b.'s too.

Lemma 4.14. (Representation lemma) For any $h \in (\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_{\perp}$ and any $d \in \mathcal{U}_{\mathcal{B}}$, we have $h(\!(d)\!) = G(h) \cdot d$.

Proof. Let $h \in (\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_{\perp}$ and $d \in \mathcal{U}_{\mathcal{B}}$. If $h = \perp$, then

$$G(h) \cdot d \quad = \quad \uparrow \omega \cdot d \quad = \quad \uparrow \omega \quad = \quad \perp(\!(d)\!)\,.$$

If $h = up(g)$, we have

$$h(\!(d)\!) = g(\bigsqcup_{\pi \in d} \uparrow \pi) = \bigcup_{f_{\psi\phi} \leqslant g} \bigcup_{\pi \in d} f_{\psi\phi}(\uparrow \pi) \text{ and}$$

$$G(h) \cdot d = \bigcup_{f_{\psi\phi} \leqslant g} G(up(f_{\psi\phi})) \cdot \bigsqcup_{\pi \in d} \uparrow \pi = \bigcup_{f_{\psi\phi} \leqslant g} \bigcup_{\pi \in d} G(up(f_{\psi\phi})) \cdot \uparrow \pi.$$

Therefore, it is enough to prove that, for any ϕ , ψ and π,

$$(*) \qquad G(up(f_{\psi\phi})) \cdot \uparrow \pi = f_{\psi\phi}(\uparrow \pi).$$

It is easily seen that $G(up(f_{\psi\phi})) = \uparrow(\psi \to \phi)$. Then we can establish (*) as follows:

$$\begin{array}{lll} & \uparrow(\psi \to \phi) \cdot \uparrow \pi & \\ = & \{\sigma \mid \psi \to \phi \leqslant \pi \to \sigma\} & \text{(by Corollary 4.12)} \\ = & \{\sigma \mid \sigma = \omega \ \text{ or } \ (\sigma \neq \omega \text{ and } \pi \leqslant \psi \text{ and } \phi \leqslant \sigma)\} & \text{(by the arrow property)} \\ = & f_{\psi\phi}(\uparrow \pi). & \end{array}$$

□

Theorem 4.15. (Solution of the domain equation) Let F and G be as in Definition 4.13.

1 $G \circ F = \mathbf{I}_{\mathcal{U}}$.
2 $F \circ G = \mathbf{I}_{(\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_{\perp}}$.

Proof.

1. Let $x \in \mathcal{U}$. If $x = \uparrow \omega$, then $G(F(\uparrow \omega)) = G(\perp) = \uparrow \omega$. Otherwise,

$$G(F(x)) = \uparrow \{\pi \to \phi \mid \phi \in x \cdot \uparrow \pi\} = \uparrow \{\pi \to \phi \mid \pi \to \phi \in x\} = x.$$

2. Let $f \in (\mathcal{U}_{\mathcal{B}} \to \mathcal{U})_{\perp}$; for any $d \in \mathcal{U}_{\mathcal{B}}$ we have

$$F(G(f))(\!(d)\!) = G(f) \cdot d = f(\!(d)\!)$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

by the representation Lemma 4.14. Hence $down(F(G(f)) = down(f)$, and $F \circ G = \mathbf{I}_{(\mathscr{U}_{\mathscr{B}} \to \mathscr{U})_\perp}$ then follows easily from the observation that $\omega \to \omega \notin \uparrow \omega$, and hence that $G(g) \neq \perp$ if $g \neq \perp$.

□

Definition 4.16. (Denotational semantics) The denotational preorder $\sqsubseteq_{\mathscr{U}}$, with associated equivalence $\simeq_{\mathscr{U}}$, is given by

$$M \sqsubseteq_{\mathscr{U}} N \overset{def}{\iff} \forall \rho \in \mathrm{CEnv}\ \mathscr{V}[\![M]\!]_\rho \subseteq \mathscr{V}[\![N]\!]_\rho .$$

5. Type assignment system

We show that the interpretation of a term in model $\mathscr{U}$ coincides with the set of types that can be assigned to it in the system $\mathscr{P}$ described below. On the way, we prove some syntactic properties of the type system such as subject expansion (Theorem 5.10) and a property of extensionality (Lemma 5.11).

5.1. *Definition and properties*

The functionality theory associated with the λ-calculus of resources with convergence testing is a sequent calculus extending the one for λ_r given in Boudol (1993), which incorporates the entailment relation $\phi \leqslant \sigma$ between types†. The provable judgments are of the form $\Gamma \vdash T : \tau$, where Γ, called a typing context, is a finite sequence of hypotheses $x_1 : \pi_1, \ldots, x_n : \pi_n$, T is a term (respectively, bag of terms) and τ is a type in Ft (respectively, Fb). We will often write $x \in \Gamma$ for $\exists \pi\ \ x : \pi \in \Gamma$. The usual structural rules are unified in a single rule allowing us to do a finite number of exchanges, weakenings and products of hypotheses.

Definition 5.1. (Type system $\mathscr{P}$) Let $\gg$ be the least reflexive, transitive relation containing the pairs

(exchange)	$\Gamma, x : \pi, y : \psi, \Delta$	$\gg$	$\Gamma, y : \psi, x : \pi, \Delta$
(weakening)	Γ	$\gg$	$x : \pi, \Gamma$
(product)	$\Gamma, x : \pi, x : \psi, \Delta$	$\gg$	$\Gamma, x : \pi \times \psi, \Delta$.

The rules of system $\mathscr{P}$ are shown in Figure 6. They are essentially those of classical intersection type systems (Sallé 1978; Coppo and Dezani 1980). The difference is that the manipulation of the hypotheses in $\mathscr{P}$ has a 'multiplicative character': contractions are not authorized. This allows us to treat independently each occurrence of a variable and to associate one resource to each one. Notice that (L1) combined with (L8) gives the more usual typing rule $\Gamma \vdash x : \phi$ if $x : \phi$ is in Γ.

† The functionality theory studied in Boudol (1993) only involves a notion of equivalence between types; entailment is not needed when discussing adequacy. Its role in the proof of full abstraction can be seen in the proof of the convergence Lemma 6.13.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Remark 5.2. Since terms are also bags, there are provable judgments $\Gamma \vdash M : \pi$, with $\pi \in \mathrm{Fb}$, but it is straightforward to see that we must then have $\pi \sim \phi$ for some $\phi \in \mathrm{Ft}$, and $\Gamma \vdash M : \phi$ (with a shorter proof). This is because the only freedom we have to apply rule (L5) is via structural equivalences of the sort $P \equiv (\mathbf{1} \mid P)$ (*cf.* Figure 2), and because the only possible types for $\mathbf{1}$ are the types equivalent to ω (through Rule (L7)).

(L1) $x : \phi \vdash x : \phi$

(L2) $$\frac{x : \pi, \Gamma \vdash N : \phi}{\Gamma \vdash \lambda x.N : \pi \to \phi} \ (x \notin \Gamma)$$

(L3) $$\frac{\Gamma \vdash M : \pi \to \phi \quad \Delta \vdash P : \pi}{\Gamma, \Delta \vdash (MP) : \phi}$$

(L4) $$\frac{\Gamma \vdash P : \pi \quad x : \pi, \Delta \vdash M : \phi}{\Gamma, \Delta \vdash M\langle P/x\rangle : \phi} \ (x \notin \Delta)$$

(L5) $$\frac{\Gamma \vdash P : \pi \quad \Delta \vdash Q : \psi}{\Gamma, \Delta \vdash (P|Q) : \pi \times \psi}$$

(L6) $$\frac{\Gamma \vdash (M|M^{\infty}) : \pi}{\Gamma \vdash M^{\infty} : \pi}$$

(L7) $\Gamma \vdash T : \omega$

(L8) $$\frac{\Delta \vdash T : \tau}{\Gamma \vdash T : \tau} \ (\Delta \gg \Gamma)$$

(L9) $$\frac{\Gamma \vdash T : \tau}{\Gamma \vdash T : \sigma} \ (\tau \leqslant \sigma)$$

(L10) $$\frac{\Gamma \vdash P : \omega \to \omega}{\Gamma \vdash \mathrm{c}P : \phi \to \phi}$$

Fig. 6. Type assignment system $\mathscr{P}$

Definition 5.3. (Type semantics) The type preorder $\sqsubseteq_{\mathscr{P}}$, with associated equivalence $\simeq_{\mathscr{P}}$, is given by

$$(M \sqsubseteq_{\mathscr{P}} N) \ \stackrel{def}{\Longleftrightarrow} \ \forall \Gamma\, \forall \phi \ (\Gamma \vdash M : \phi \ \Rightarrow \ \Gamma \vdash N : \phi)\,.$$

Lemma 5.4. The following rule is admissible in system $\mathscr{P}$:

$$\frac{x : \pi, \Gamma \vdash T : \tau}{x : \psi, \Gamma \vdash T : \tau} \ (\psi \leqslant \pi)$$

that is, if the assumption is provable by the rules of $\mathscr{P}$, then so is the conclusion.

Proof. The proof is by structural induction on the proof of $x : \pi, \Gamma \vdash T : \tau$. The interesting case is that of Axiom (L1). We show the implication $\psi \leqslant \phi \Rightarrow x : \psi \vdash x : \phi$ as follows. Let $\psi = \sigma_1 \times \cdots \times \sigma_n \leqslant \phi$. Then by the product property, there exists i such

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

that $\sigma_i \leqslant \phi$. Then

$$\dfrac{\dfrac{\dfrac{x:\sigma_i \vdash x:\sigma_i}{x:\sigma_i \vdash x:\phi}\,(L9)}{x:\sigma_1,\ldots,x:\sigma_n \vdash x:\phi}\,(L8)}{x:\psi \vdash x:\phi}\,(L8)\,.$$

□

Lemma 5.4 allows us to perform additional weakenings during proofs. From now on, we consider relation $\gg$ enriched with the weakening axiom

$$\psi \leqslant \pi \quad \Rightarrow \quad (x:\pi,\Gamma \gg x:\psi,\Gamma).$$

That is, we incorporate Lemma 5.4 into the rule (L8).

Definition 5.5.

— We shall freely manipulate contexts as multisets, that is, up to uses of exchange.
— Let T be a term or a bag of terms and y a variable. We then define Γ_T and $\Gamma/_y$ by induction on Γ as follows:

$$\Gamma \text{ empty} \quad \Rightarrow \quad \left[\begin{array}{ll} \Gamma_T & \text{empty} \\ \Gamma/_y & \text{empty} \end{array}\right.$$

$$\Gamma = x:\pi,\Delta \quad \Rightarrow \quad \left[\begin{array}{lll} \Gamma_T & = & \begin{cases} x:\pi,\Delta_T & \text{if } x \in fv(T) \\ \Delta_T & \text{otherwise} \end{cases} \\ & & \\ \Gamma/_y & = & \begin{cases} x:\pi,\Delta/_y & \text{if } x \neq y \\ \Delta/_y & \text{otherwise.} \end{cases} \end{array}\right.$$

— If $\Gamma_x = x:\pi$ (there is only one occurrence of x in Γ), then we set $\Gamma(x) = \pi$.
— We use $\Gamma^\times$ to denote the context such that if $\Gamma_x = x:\pi_1,\ldots,x:\pi_n$, then $\Gamma^\times(x) = \pi_1 \times \cdots \times \pi_n$ (for any x).
— We use $\tilde{\Gamma}$ to denote the smallest typing context (defined up to exchange) such that for any $x:\pi \in \Gamma$, if $\pi = \phi_1 \times \cdots \times \phi_n$ with $\phi_i \in \mathrm{Ft}$, then $x:\phi_1,\ldots,x:\phi_n \in \tilde{\Gamma}$.

Next we state some simple properties of system $\mathscr{P}$ in the following proposition.

Proposition 5.6.

1 $\Gamma \vdash T:\tau \;\Rightarrow\; \tilde{\Gamma} \vdash T:\tau$.
2 $\Gamma^\times \vdash T:\tau \;\Rightarrow\; \Gamma \vdash T:\tau$.
3 $\Gamma \vdash T:\tau$ and $y \notin fv(T) \;\Rightarrow\; \Gamma/_y \vdash T:\tau$. So $\Gamma_T \vdash T:\tau$ holds too.
4 $\Gamma \gg \Delta \;\Rightarrow\; \Delta^\times_x(x) \leqslant \Gamma^\times_x(x)$ (for all x).
5 $\Delta \gg \Gamma$ and $x:\psi \in \Delta \;\Rightarrow\; x:\pi \in \Gamma$ and $\pi \leqslant \psi$ (for some π).

Proof. 1. We use structural induction on the derivation $\Gamma \vdash T:\tau$. Besides the induction hypotheses, we use the following properties: $x \mathrel{\tilde{:}} \pi, \tilde{\Gamma} \vdash M:\sigma$ implies $x:\pi,\tilde{\Gamma} \vdash M:\sigma$ by (L8); $\Gamma = \Gamma_1,\Gamma_2$ implies $\tilde{\Gamma} = \tilde{\Gamma}_1,\tilde{\Gamma}_2$; $\Delta \gg \Gamma$ implies $\tilde{\Delta} \gg \tilde{\Gamma}$.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

2. Assume $\Gamma^\times \vdash M : \phi$. Then, from Part (1) of this proposition, $\tilde{\Gamma^\times} \vdash M : \phi$ holds. Furthermore, $\tilde{\Gamma^\times} \gg \tilde{\Gamma} \gg \Gamma$. By (L8), we get $\Gamma \vdash M : \phi$.

3. The proof is a straightforward induction on the derivation of $\Gamma \vdash T : \tau$.

4. We use induction on the size of the proof of $\Gamma \gg \Delta$. We examine weakening only. Let $\Gamma \gg y : \theta, \Gamma = \Delta$. If $y \neq x$, then $\Gamma_x = \Delta_x$. If $y = x$, then $\Delta_x^\times(x) = \theta \times \Gamma_x^\times(x) \leqslant \omega \times \Gamma_x^\times(x) \leqslant \Gamma_x^\times(x)$.

5. The proof is similar to that for Part (4). □

The following proposition states that type assignment within $\mathscr{P}$ is syntax-directed up to uses of rules (L8) and (L9).

Proposition 5.7.

1 If $\Gamma \vdash P : \pi$ and $\pi \triangleright \phi_1, \ldots, \phi_n$ with $n \geqslant 1$, then there exist terms $M_1, \ldots, M_n, Q$ and contexts $\Gamma_1, \ldots \Gamma_n$ such that $P \equiv (M_1 \mid \cdots \mid M_n \mid Q)$, $\Gamma_1, \ldots, \Gamma_n \gg \Gamma$ and $\forall i \in \{1, \ldots, n\}\ \Gamma_i \vdash M_i : \phi_i$ (with shorter proofs).

2 If $\Gamma \vdash M\langle P/x \rangle : \phi$ and $\phi \neq \omega$, there exist ψ, Γ_1, Γ_2 such that

$$\Gamma_1 \vdash P : \psi \quad \text{and} \quad x : \psi, \Gamma_2 \vdash M : \phi \quad \text{where} \quad x \notin \Gamma_2 \quad \text{and} \quad \Gamma_1, \Gamma_2 \gg \Gamma.$$

3 If $\Gamma \vdash MP : \phi$ and $\phi \neq \omega$, there exist π, Γ_1, Γ_2 such that

$$\Gamma_1 \vdash M : \pi \to \phi \quad \text{and} \quad \Gamma_2 \vdash P : \pi \quad \text{where} \quad \Gamma_1, \Gamma_2 \gg \Gamma.$$

4 If $\Gamma \vdash \lambda x.M : \phi$ and $\phi \neq \omega$, then there exist π, σ, Δ such that

$$x : \pi, \Delta \vdash M : \sigma \quad \text{where} \quad \phi = \pi \to \sigma,\ x \notin \Delta \quad \text{and} \quad \Delta \gg \Gamma.$$

Proof. All proofs are by structural induction. We examine one case of Part 1 and one case of Part 3 of the statement:

1. Suppose that $\Gamma \vdash P : \pi$ is derived by (L9). Then $\Gamma \vdash P : \psi$ with $\psi \leqslant \pi$. By induction, $\psi \triangleright \delta_1, \ldots, \delta_r$, $P \equiv (N_1 \mid \cdots \mid N_r \mid Q')$, $\Gamma_i \vdash N_i : \delta_i$ and $\Gamma_1, \ldots, \Gamma_r \gg \Gamma$. By the product property, there exists an injection $b : [1, n] \to [1, r]$ such that $\delta_{b(j)} \leqslant \phi_j$ for any $j = 1, \ldots, n$. This implies $n \leqslant r$ and $\Gamma_{b(j)} \vdash N_{b(j)} : \phi_j$ for any $j = 1, \ldots, n$. Then, writing

$$(N_1 \mid \cdots \mid N_r) \equiv (N_{b(1)} \mid \cdots \mid N_{b(n)} \mid Q'') \text{ and } Q = (Q'' \mid Q'),$$

we have $P \equiv (N_{b(1)} \mid \cdots \mid N_{b(n)} \mid Q)$. Thus the conclusion holds with $M_j = N_{b(j)}$.

3. If the derivation of $\Gamma \vdash MP : \phi$ ends with (L9), then $\Gamma \vdash MP : \sigma$ and $\sigma \leqslant \phi$. By induction, $\Gamma_1 \vdash M : \pi \to \sigma$ and $\Gamma_2 \vdash P : \pi$ where $\Gamma_1, \Gamma_2 \gg \Gamma$. Since $\sigma \leqslant \phi$ implies $\pi \to \sigma \leqslant \pi \to \phi$, we have $\Gamma_1 \vdash M : \pi \to \phi$ by Rule (L9). □

Proposition 5.8.

1 $\Gamma \vdash x : \phi \;\Rightarrow\; (\phi \neq \omega \;\Rightarrow\; \exists x : \pi \in \Gamma\ \pi \leqslant \phi)$.
2 $\Gamma \vdash \mathbf{I} : \pi \to \phi \;\Rightarrow\; \pi \leqslant \phi$.
3 $\Gamma \vdash \mathrm{c}x : \psi \to \sigma \;\Rightarrow\; \exists x : \pi \in \Gamma\ \pi \leqslant \omega \to \omega$ and $\psi \leqslant \sigma$.
4 $\Gamma \vdash x^\infty : \pi \;\Rightarrow\; (\pi \nsim \omega \;\Rightarrow\; \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma\ \psi_1 \times \cdots \times \psi_n \leqslant \pi)$.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Proof. Part (4) is an easy consequence of proposition 5.7 (1) and of Part (1) of this statement. Parts (1), (2) and (3) are proved by structural induction. We examine a few cases.

1. If the derivation ends with (L8), we have $\Delta \vdash x : \phi$ for some Δ such that $\Delta \gg \Gamma$. By induction, some $x : \psi$ exists in Δ with $\psi \leqslant \phi$. By Proposition 5.6 (5), $x : \pi \in \Gamma$ for some π such that $\pi \leqslant \psi$. Hence, $\pi \leqslant \phi$ holds by transitivity.

2. We can assume $\phi \neq \omega$, since otherwise $\pi \leqslant \omega$ for any π. Assume the last rule used in the derivation is (L2). Then $x : \pi, \Gamma \vdash x : \phi$ with $x \notin \Gamma$. By Part (1) of this proposition, we have $\pi \leqslant \phi$. If the last rule is (L9), then $\Gamma \vdash \mathbf{I} : \delta$ for some $\delta \leqslant \pi \to \phi$. Since $\delta \in \mathrm{Ft}$, we must have $\delta = \pi' \to \phi'$ for some formulas π' and ϕ'. As $\phi \neq \omega$, the arrow property guarantees $\phi' \neq \omega$, $\pi \leqslant \pi'$ and $\phi' \leqslant \phi$. Moreover, $\pi' \leqslant \phi'$ holds by induction, hence $\pi \leqslant \phi$ by transitivity.

3. Assume rule (L10) is used last. Then $\psi = \sigma$ and $\Gamma \vdash x : \omega \to \omega$. By Part (1) of this proposition, there exists $x : \pi$ in Γ such that $\pi \leqslant \omega \to \omega$. □

Proposition 5.9.

1 $M =_\alpha M'$ and $\Gamma \vdash M : \phi \Rightarrow \Gamma \vdash M' : \phi$.
2 $P \equiv P'$ and $\Gamma \vdash P : \pi \Rightarrow \Gamma \vdash P' : \pi$.
3 $P \propto Q$ and $\Gamma \vdash Q : \pi \Rightarrow \Gamma \vdash P : \pi$. In particular, $x : \pi \vdash x^\infty : \pi$ since $x^\infty \propto x$ and $x : \pi \vdash x : \pi$.

Proof. Part (1) is a consequence of the following property, which is established by a straightforward induction: if variables are consistently renamed in the context and in the term, then the resulting judgment is still provable. Part (2) follows from the observation that Rule (L6) is in fact reversible (applying Rule (L5) and assigning type ω to M). Also, the only types for $\mathbf{1}$ are those equivalent to ω. As for Part (3), $P \propto Q$ implies $P \equiv (Q \mid P')$, hence the statement holds by Part (2) (assigning type ω to P'). □

The evaluation relation $\to_{rc}$ and the typing system $\mathscr{P}$ verify the important property of *subject expansion*: the types of the terms obtained during an evaluation of M are types of M too.

Theorem 5.10. (Subject expansion) If $T \to_{rc} T'$ and $\Gamma \vdash T' : \tau$, then $\Gamma \vdash T : \tau$.

Proof. We shall use the following property, which is easily checked:

$$(*) \quad (M\langle N/x\rangle \succ M' \text{ and } \Gamma \vdash M' : \phi) \quad \Rightarrow \quad \Gamma \vdash M\langle N/x\rangle : \phi,$$

The proof of the lemma is by induction on the derivation of $T \to_{rc} T'$. We examine a few cases:

— If $T = (\lambda x.M)P \to_{rc} M\langle P/x\rangle = T'$, then $\Gamma_1 \vdash P : \psi$ and $x : \psi, \Gamma_2 \vdash M : \tau$ with $\Gamma_1, \Gamma_2 \gg \Gamma$ and $x \notin \Gamma_2$ by Proposition 5.7 (2). Then $\Gamma \vdash (\lambda x.M)P : \tau$ by (L2) and (L3).

— If $T = \mathrm{c}V \to_{rc} \mathbf{I} = T'$, then $\phi = \pi \to \sigma$ and $\pi \leqslant \sigma$ hold by Proposition 5.8 (2). On the other side, $\Gamma \vdash \mathrm{c}V : \sigma \to \sigma$ holds for any value V, by (L10). So $\Gamma \vdash T : \phi$ follows by (L9), since $\sigma \to \sigma \leqslant \phi$.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

— If $T = M\langle P/x\rangle \rightarrow_{rc} M'\langle Q/x\rangle = T'$, where $P \equiv (N \mid Q)$, $M\langle N/x\rangle \succ M'$ and $x \notin fv(N)$, then

$$\Gamma_1 \vdash Q : \psi \quad \text{and} \quad x : \psi, \Gamma_2 \vdash M' : \phi$$

with $x \notin \Gamma_2$ and $\Gamma_1, \Gamma_2 \gg \Gamma$ by Proposition 5.7 (2). Then, $x : \psi, \Gamma_2 \vdash M\langle N/x\rangle : \phi$ by the property (∗). Since x is bound in $M\langle N/x\rangle$, we have, in fact, $\Gamma_2 \vdash M\langle N/x\rangle : \phi$, by proposition 5.6 (3). By Proposition 5.7 (2) again, there exist π, Σ', Σ'' such that

$$\Sigma' \vdash N : \pi \quad \text{and} \quad x : \pi, \Sigma'' \vdash M : \phi,$$

with $x \notin \Sigma''$ and $\Sigma', \Sigma'' \gg \Gamma_2$. Since $x : \pi, \Sigma'' \gg x : \pi, x : \psi, \Sigma'' \gg x : \pi \times \psi, \Sigma''$, we get $x : \pi \times \psi, \Sigma'' \vdash M : \phi$ by (L8). Then $\Sigma', \Gamma_1, \Sigma'' \vdash M\langle P/x\rangle : \phi$ by (L4), since $\Sigma', \Gamma_1 \vdash (N \mid Q) : \pi \times \psi$ by (L5). This implies $\Gamma \vdash T : \phi$ by (L8). □

We next prove an extensionality lemma for our typing system. In the usual intersection type systems, this property is expressed as follows (*cf.* Hindley (1982), Hindley (1983) and Dezani and Margaria (1986)):

$$(\Gamma \vdash M : \phi \wedge (\omega \rightarrow \omega) \text{ and } x \notin fv(M)) \quad \Rightarrow \quad \Gamma \vdash \lambda x.Mx : \phi \,.$$

Lemma 5.11. (Extensionality lemma) Let $x \notin fv(M)$, $x \notin \Gamma$. The following statement holds:

$$(\Gamma \vdash M : \pi \rightarrow \sigma) \iff (\Gamma \vdash M : \omega \rightarrow \omega \text{ and } x : \pi, \Gamma \vdash Mx^{\infty} : \sigma)\,.$$

Proof. (⇒) If $\Gamma \vdash M : \pi \rightarrow \sigma$, we have $\Gamma \vdash M : \omega \rightarrow \omega$ by (L9). By Proposition 5.9 (3), for any π we have $x : \pi \vdash x^{\infty} : \pi$, then $x : \pi, \Gamma \vdash Mx^{\infty} : \sigma$ by (L3).

(⇐) Assume $\Gamma \vdash M : \omega \rightarrow \omega$ and $x : \pi, \Gamma \vdash Mx^{\infty} : \sigma$ with $x \notin \Gamma$. If $\sigma = \omega$, we know $\omega \rightarrow \omega \leqslant \pi \rightarrow \omega \leqslant \pi \rightarrow \sigma$, so $\Gamma \vdash M : \pi \rightarrow \sigma$ by (L9). Let $\sigma \neq \omega$. We prove $\Gamma \vdash M : \pi \rightarrow \sigma$ by structural induction on the derivation of $x : \pi, \Gamma \vdash Mx^{\infty} : \sigma$, up to uses of (L8). One distinguishes cases according to which is the last rule applied. We will examine the case (L3) only. Let $\Sigma \vdash M : \psi \rightarrow \sigma$ and $\Delta \vdash x^{\infty} : \psi$, where $\Sigma, \Delta = x : \pi, \Gamma$. If $\psi \not\sim \omega$, then $\Delta_x = x : \pi$ and $\pi \leqslant \psi$ by Proposition 5.8 (4). If $\psi \sim \omega$, we obviously also have $\pi \leqslant \psi$. On the other hand, since $\Sigma \gg x : \pi, \Gamma$, we have $x : \pi, \Gamma \vdash M : \psi \rightarrow \sigma$ by (L8), and $\Gamma \vdash M : \psi \rightarrow \sigma$ since $x \notin fv(M)$. The conclusion then follows by (L9). □

5.2. *Equivalence between the model and the type system*

In this section, we show that the meaning of a term in the model $\mathscr{U}$ corresponds exactly to the set of types that can be given to this term in system $\mathscr{P}$. We specialize the semantic function to the logical model $\mathscr{U}$. Notice that ξ is a compact environment if and only if for any variable x in its domain, $\xi(x) = \uparrow \pi$ for some $\pi \in$ Fb. A consequence of the isomorphism between $\mathscr{U}_{\mathscr{B}}$ and $\mathscr{M}(\mathscr{U})$ is that the product of compact environments becomes

$$(\forall i\ \forall x\ \rho_i(x) = \uparrow \pi_i) \Rightarrow (\rho_m \centerdot \cdots \centerdot \rho_n)(x) = \uparrow (\pi_1 \times \cdots \times \pi_n)\,.$$

Because $\mathscr{U}$ solves the domain equation up to isomorphism, the equations defining the meaning of terms now involve functions F and G in the standard way. The interpretation function $\mathscr{V} : \Lambda_{rc} \times \text{Env} \rightarrow \mathscr{U}$ is defined in Figure 7.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

$$\mathcal{V}[\![x]\!]_\rho = \{\phi \in \mathrm{Ft} \mid \phi \in \rho(x)\}$$

$$\mathcal{V}[\![\lambda x.M]\!]_\rho = \uparrow\{\pi \to \phi \mid \phi \in \mathcal{V}[\![M]\!]_{\rho[x:=\uparrow\pi]}\}$$

$$\mathcal{V}[\![MP]\!]_\rho = \{\sigma \mid \pi \in \uparrow \phi_1 \times \cdots \times \phi_n \text{ and } (\pi \to \sigma) \in \mathcal{V}[\![M]\!]_{\rho_0}\} \cup \{\omega\} \quad \text{where } \begin{cases} \rho \supseteq \rho_0 \centerdot \rho_1 \centerdot \cdots \centerdot \rho_n \\ P \propto (M_1 \mid \cdots \mid M_n) \\ \phi_i \in \mathcal{V}[\![M_i]\!]_{\rho_i} \end{cases}$$

$$\mathcal{V}[\![M\langle P/x\rangle]\!]_\rho = \bigcup \mathcal{V}[\![M]\!]_{\rho_0[x:=\uparrow\phi_1\times\cdots\times\phi_n]} \quad \text{where } \begin{cases} \rho \supseteq \rho_0/_x \centerdot \rho_1 \centerdot \cdots \centerdot \rho_n \\ P \propto (M_1 \mid \cdots \mid M_n) \\ \phi_i \in \mathcal{V}[\![M_i]\!]_{\rho_i} \end{cases}$$

$$\mathcal{V}[\![\mathrm{c}P]\!]_\rho = \begin{cases} \mathcal{V}[\![\mathbf{I}]\!] & \text{if } P \equiv (M \mid Q) \text{ and } \omega \to \omega \in \mathcal{V}[\![M]\!]_\rho \\ \bot & \text{otherwise} \end{cases}$$

Fig. 7. Interpretation function over $\mathcal{U}$

Lemma 5.12. Let $M \in \Lambda_{rc}$ and $\rho, \rho' \in \mathrm{Env}$. The following properties hold:

— Monotonicity: $\rho \subseteq \rho' \Rightarrow \mathcal{V}[\![M]\!]_\rho \sqsubseteq \mathcal{V}[\![M]\!]_{\rho'}$.
— Continuity: $\tau \in \mathcal{V}[\![M]\!]_\rho \iff \exists \xi \in \mathrm{CEnv}\ (\xi \subseteq \rho \text{ and } \tau \in \mathcal{V}[\![M]\!]_\xi)$.

Proof. The proof is a straightforward induction on the definition of the interpretation function. □

Lemma 5.13. The preorder $\sqsubseteq_{\mathscr{P}}$ (*cf.* Definition 5.3) is a precongruence and is closed with respect to $=_\alpha$ and $\equiv$:

— $M \sqsubseteq_{\mathscr{P}} N \Rightarrow \forall C\ C[M] \sqsubseteq_{\mathscr{P}} C[N]$.
— $(M =_\alpha N \text{ or } M \equiv N) \Rightarrow (M \simeq_{\mathscr{P}} N)$.

Proof. Part (1) is shown by induction on the context C, using Proposition 5.7. Part (2) holds by Proposition 5.9. □

Definition 5.14.

— Given Γ, the compact environment ρ_Γ is defined by

$$\rho_\Gamma(x) = \uparrow \Gamma^\times(x) \quad \text{for all variables } x \text{ in } \Gamma\ .$$

— Given $\xi \in \mathrm{CEnv}$, the context Γ_ξ is defined by

$$\Gamma_\xi = \{x : \pi \mid \xi(x) = \uparrow \pi\}\ .$$

Lemma 5.15. For any Δ, Γ and compact environment ξ, the following properties hold:

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

1 $\Delta \gg \Gamma \iff \rho_\Delta(x) \subseteq \rho_\Gamma(x)$ for any x in the domain of Δ.
2 $\rho_{\Delta,\Gamma} = \rho_\Delta \centerdot \rho_\Gamma$.
3 $\rho_{\Gamma_\xi} = \xi$.

Proof. 1. This follows from Proposition 5.6.

2. By definition, $\rho_{\Delta,\Gamma}(x) = \uparrow (\Delta,\Gamma)^\times(x) = \uparrow (\Delta^\times(x) \times \Gamma^\times(x)) = \rho_\Delta \centerdot \rho_\Gamma$.

3. We have

$$\begin{array}{rcll} \rho_{\Gamma_\xi}(x) & = & \uparrow \Gamma_\xi^\times(x) & \text{for any } x \in \Gamma_\xi \\ & = & \uparrow \Gamma_\xi(x) & \text{since } \xi \text{ contains only one assumption on } x \\ & = & \uparrow \pi & \text{with } \xi(x) = \uparrow \pi\,. \end{array}$$

□

Lemma 5.16. For all Γ, N, τ: $(\Gamma \vdash N : \tau) \;\Rightarrow\; \tau \in \mathscr{V}[\![N]\!]_{\rho_\Gamma}$.

Proof. For $\tau = \omega$, the statement holds immediately. Assume $\tau \neq \omega$. We proceed by structural induction on the derivation of $\Gamma \vdash N : \tau$.

(L1) Here $\Gamma = x : \tau$ and $N = x$. Since $\mathscr{V}[\![x]\!]_{\rho_\Gamma} = \uparrow \tau$, we have that $\tau \in \mathscr{V}[\![x]\!]_{\rho_\Gamma}$ holds.

(L2) Here $N = \lambda x.M$ and $\tau = \pi \to \phi$. Let $\Delta = x : \pi, \Gamma$. By induction, we have $\phi \in \mathscr{V}[\![M]\!]_{\rho_\Delta}$. Moreover, $\rho_\Delta = \rho_\Gamma[x := \uparrow \pi]$, so $\pi \to \phi \in \mathscr{V}[\![N]\!]_{\rho_\Gamma}$ by definition of the semantic function.

(L3) Here $N = (MP)$. Assume $\Delta \vdash P : \pi$ and $\Sigma \vdash M : \pi \to \phi$, with $\Gamma = \Sigma, \Delta$. By induction, $\pi \to \phi \in \mathscr{V}[\![M]\!]_{\rho_\Sigma}$. Consider the case $\pi \not\sim \omega$. Let $\pi \triangleright \phi_1, \ldots, \phi_n$. Then there exist $M_1, \ldots, M_n, R$ and $\Delta_1, \ldots, \Delta_n$ such that $P \equiv (M_1 \mid \cdots \mid M_n \mid R)$ and $\Delta_i \vdash M_i : \phi_i$ with $\Delta_1, \ldots, \Delta_n \gg \Delta$. By the induction hypothesis, we have $\phi_i \in \mathscr{V}[\![M_i]\!]_{\rho_{\Delta_i}}$ for any i. Since $\Sigma, \Delta_1, \ldots, \Delta_n \gg \Gamma$ implies $\rho_\Gamma \supseteq \rho_\Sigma \centerdot \rho_{\Delta_1} \centerdot \cdots \centerdot \rho_{\Delta_n}$ by Lemma 5.15, and since $P \propto (M_1 \mid \cdots \mid M_n)$, we have $\phi \in \mathscr{V}[\![MP]\!]_{\rho_\Gamma}$.

(L4) Here $N = M\langle P/x\rangle$. Assume $x : \pi, \Sigma \vdash M : \tau$ and $\Delta \vdash P : \pi$, with $\Sigma, \Delta \gg \Gamma$ and $x \notin \Sigma$. We have $\tau \in \mathscr{V}[\![M]\!]_{\rho_{\Sigma[x:=\uparrow\pi]}}$ by induction. Let $\pi \triangleright \phi_1, \ldots, \phi_n$. There exist some $M_1, \ldots, M_n$ and $\Delta_1, \ldots, \Delta_n$ such that $P \propto (M_1 \mid \cdots \mid M_n)$ and $\Delta_i \vdash M_i : \phi_i$ with $\Delta_1, \ldots, \Delta_n \gg \Delta$. Then $\phi_i \in \mathscr{V}[\![M_i]\!]_{\rho_{\Delta_i}}$ holds by induction. Since $\xi_\Gamma \supseteq \rho_\Sigma \centerdot \rho_{\Delta_1} \centerdot \cdots \centerdot \rho_{\Delta_n}$ and $\pi \in \uparrow \pi$, we have $\tau \in \mathscr{V}[\![M\langle P/x\rangle]\!]_{\rho_\Gamma}$.

(L8) This follows from Lemma 5.15 (1) and by monotonicity (Lemma 5.12).

(L10) Here $N = \mathrm{c}P$, $\tau = \phi \to \phi$ and $\Gamma \vdash P : \omega \to \omega$. By Proposition 5.7, there exist M, Q, Γ' such that $P \equiv (M \mid Q)$ and $\Gamma' \vdash M : \omega \to \omega$, where $\Gamma' \gg \Gamma$. By induction, $\omega \to \omega \in \mathscr{V}[\![M]\!]_{\rho_{\Gamma'}}$. By Lemma 5.15 and by monotonicity, we have $\omega \to \omega \in \mathscr{V}[\![M]\!]_{\rho_\Gamma}$. Then $\phi \to \phi \in \mathscr{V}[\![\mathbf{I}]\!] = \mathscr{V}[\![\mathrm{c}P]\!]_{\rho_\Gamma}$. □

Lemma 5.17. Let ξ be a compact environment; the meaning of a term N in this environment is obtained from the typing system as follows:

$$\mathscr{V}[\![N]\!]_\xi = \{\tau \mid \Gamma_\xi \vdash N : \tau\}\,.$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Proof. Let $\Gamma_\xi \vdash N : \tau$. Then $\tau \in \mathscr{V}[\![N]\!]_{\rho_{\Gamma_\xi}} = \mathscr{V}[\![N]\!]_\xi$ by Lemmas 5.16 and 5.15. We prove the other inclusion by structural induction on N. For $\tau = \omega$, the lemma holds immediately by Rules (L7) and (L9). Assume $\tau \neq \omega$ and $\tau \in \mathscr{V}[\![N]\!]_\xi$.

$N = x$: Let $\xi(x) = \uparrow \pi$; since $\tau \in \xi(x)$, we have $\pi \leqslant \tau$. Moreover, $x : \tau \vdash x : \tau$ and $x : \tau \gg x : \pi \gg \Gamma_\xi$, hence we obtain $\Gamma_\xi \vdash x : \tau$ by (L8).

$N = \lambda x.M$: Let $\tau = \pi \to \phi \in \mathscr{V}[\![\lambda x.M]\!]_\xi$. Hence there exists $\psi \to \sigma$ such that

$$\sigma \in \mathscr{V}[\![M]\!]_{\xi[x:=\uparrow\psi]} \text{ and } \psi \to \sigma \leqslant \pi \to \phi .$$

By the arrow property, we have either $\phi = \omega$, in which case $\phi \in \mathscr{V}[\![M]\!]_{\xi[x:=\uparrow\psi]}$ since denotations are non-empty, or $\sigma \neq \omega$, $\phi \neq \omega$, $\pi \leqslant \psi$ and $\sigma \leqslant \phi$, and hence $\phi \in \mathscr{V}[\![M]\!]_{\xi[x:=\uparrow\psi]}$. Since $\uparrow \psi \subseteq \uparrow \pi$, we have $\phi \in \mathscr{V}[\![M]\!]_{\xi[x:=\uparrow\pi]}$ by monotonicity. Then $\Gamma_{\xi[x:=\uparrow\pi]} \vdash M : \phi$ holds by induction. Finally, $\Gamma_{\xi[x:=\uparrow\pi]} = \Gamma_{\xi/x}, x : \pi$ and $\Gamma_{\xi/x} \gg \Gamma_\xi$ imply $\Gamma_\xi \vdash \lambda x.M : \pi \to \phi$ by (L2) and (L8).

$N = (MP)$: $\tau \in \mathscr{V}[\![MP]\!]_\xi$ means that there exist $\pi, \phi_1, \ldots, \phi_n$, $M_1, \ldots, M_n$ and compact $\xi_0, \xi_1, \ldots, \xi_n$ such that $P \propto (M_1 \mid \cdots \mid M_n)$, $\xi \sqsupseteq \xi_0 \centerdot \cdots \centerdot \xi_n$, $(\forall i\ \phi_i \in \mathscr{V}[\![M_i]\!]_{\xi_i})$, $\pi \in \uparrow \phi_1 \times \cdots \times \phi_n$, and $\pi \to \tau \in \mathscr{V}[\![M]\!]_{\xi_0}$. We have $\Gamma_{\xi_i} \vdash M_i : \phi_i$ and $\Gamma_{\xi_0} \vdash M : \pi \to \tau$ by induction. By (L5) and (L9), we get

$$\Gamma_{\xi_1}, \ldots, \Gamma_{\xi_n} \vdash (M_1 \mid \cdots \mid M_n) : \pi,$$

from which $\Gamma_{\xi_1}, \ldots, \Gamma_{\xi_n} \vdash P : \pi$ follows by (L7) since $P \equiv (M_1 \mid \cdots \mid M_n \mid Q)$ for some Q. By (L3), $\Gamma_{\xi_0}, \Gamma_{\xi_1}, \ldots, \Gamma_{\xi_n} \vdash MP : \tau$. Finally, $\xi \sqsupseteq \xi_0 \centerdot \cdots \centerdot \xi_n$ implies $\Gamma_{\xi_0}, \Gamma_{\xi_1}, \ldots, \Gamma_{\xi_n} \gg \Gamma_\xi$. Then $\Gamma_\xi \vdash MP : \tau$ follows by (L8).

$N = M\langle P/x\rangle$: With the same assumptions as for application, except that $\xi \sqsupseteq \xi'_0 \centerdot \cdots \centerdot \xi_n$ with $\xi'_0 = \xi_0/x$, we have $\tau \in \mathscr{V}[\![M]\!]_{\xi_0[x:=\uparrow\phi_1\times\cdots\times\phi_n]}$. We set $\pi = \phi_1 \times \cdots \times \phi_n$. By induction we have $\Gamma_{\xi_0[x:=\uparrow\pi]} \vdash M : \tau$ and $\Gamma_{\xi_i} \vdash M_i : \phi_i$. A derivation of $\Gamma_{\xi_1}, \ldots, \Gamma_{\xi_n} \vdash P : \pi$ is obtained as above. Since $\Gamma_{\xi_0[x:=\uparrow\pi]} = \Gamma_{\xi'_0}, x : \pi$ and $\Gamma_{\xi'_0}, \Gamma_{\xi_1}, \ldots, \Gamma_{\xi_k} \gg \Gamma_\xi$, we conclude that $\Gamma_\xi \vdash N : \tau$ by (L4) and (L8).

$N = \mathrm{c}P$: By definition, $\tau \in \mathscr{V}[\![\mathbf{I}]\!]$ and $\omega \to \omega \in \mathscr{V}[\![M]\!]_\xi$, where $P \equiv (M \mid Q)$. Since $\tau = \pi \to \sigma$ and $\sigma \in \mathscr{V}[\![x]\!]_{[x:=\uparrow\pi]}$, we have $\pi \leqslant \sigma$. On the other hand, $\Gamma_\xi \vdash M : \omega \to \omega$ holds by induction, from which $\Gamma_\xi \vdash P : \omega \to \omega$ follows, giving type ω to Q. Using (L10), we have $\Gamma_\xi \vdash \mathrm{c}P : \sigma \to \sigma$, and hence $\Gamma_\xi \vdash \mathrm{c}P : \pi \to \sigma$ holds by (L9). □

Theorem 5.18. For all $M \in \Lambda_{rc}$ and $\xi \in \mathrm{CEnv}$,

$$\mathscr{V}[\![M]\!]_\xi = \{\sigma \mid \exists \Gamma\ \Gamma \vdash M : \sigma \text{ and } \rho_\Gamma \subseteq \xi\} .$$

Proof. Assume $\Gamma \vdash M : \sigma$ and $\rho_\Gamma \subseteq \xi$. Then $\sigma \in \mathscr{V}[\![M]\!]_{\rho_\Gamma}$ by Lemma 5.16. By monotonicity of the interpretation, we have $\sigma \in \mathscr{V}[\![M]\!]_\xi$. The other inclusion is immediate by Lemma 5.17. □

Theorem 5.19. **($\mathscr{U}$ and $\mathscr{P}$ are equivalent)** For all $M, N \in \Lambda_{rc}$,

$$(M \sqsubseteq_{\mathscr{U}} N) \iff (M \sqsubseteq_{\mathscr{P}} N) .$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Proof. Let $M, N \in \Lambda_{rc}$ be such that $M \sqsubseteq_{\mathscr{U}} N$ and assume $\Gamma \vdash M : \phi$. We have

$$\begin{array}{ll} \phi \in \mathscr{V}[\![M]\!]_{\rho_\Gamma} & \text{(by Lemma 5.16)} \\ \phi \in \mathscr{V}[\![N]\!]_{\rho_\Gamma} & \text{(since } M \sqsubseteq_{\mathscr{U}} N) \\ \Gamma_{\rho_\Gamma} \vdash N : \phi & \text{(by Lemma 5.17)} \\ \tilde{\Gamma}_{\rho_\Gamma} \vdash N : \phi & \text{(by Proposition 5.6 (1))} \\ \Gamma \vdash N : \phi & \text{(since } \tilde{\Gamma}_{\rho_\Gamma} = \tilde{\Gamma} \gg \Gamma) \,. \end{array}$$

The proof of the other implication is similar. □

6. Full abstraction

The abstract semantics of λ_r^c in terms of typability within the system $\mathscr{P}$ is adequate with respect to the observational semantics we adopted for the calculus. This property extends the adequacy result proved for λ_r by Boudol (Boudol 1993). The proof technique consists in showing the soundness of typing with respect to a *realizability predicate* $\Gamma \vDash T : \tau$.

Moreover, we can establish that the abstract semantics is actually fully abstract as a consequence of the completeness result

$$\text{if } \Gamma \vDash M : \phi, \text{ then } \Gamma \vdash M : \phi.$$

The proof of completeness relies on the existence of *characteristic terms* M_σ for each type, with the property that, for any τ, M_σ has type τ if and only if $\sigma \leqslant \tau$.

6.1. *Realizability of types*

The realizability predicate $\vDash$ used here, and defined below, is the same as that given in Boudol (1993).

Definition 6.1. For closed terms and bags of λ_r^c, predicate $\vDash$ is defined by

$$\begin{array}{lcl} \vDash M : \omega & \stackrel{def}{\Longleftrightarrow} & \text{true} \\[2ex] \vDash M : \pi \to \phi & \stackrel{def}{\Longleftrightarrow} & M{\Downarrow_{rc}} \text{ and } \forall P \; (\vDash P : \pi \Rightarrow \vDash MP : \phi) \\[2ex] \vDash P : \pi & \stackrel{def}{\Longleftrightarrow} & \pi \triangleright \phi_1, \ldots, \phi_n \text{ and } n > 0 \text{ and} \\ & & \exists R, M_1, \ldots, M_n \;\; (P \equiv (M_1 \mid \cdots \mid M_n \mid R) \text{ and } \vDash M_i : \phi_i) \,. \end{array}$$

We now extend the predicate to the open terms of λ_r^c. Let $\Gamma = x_1 : \pi_1, \ldots, x_n : \pi_n$ be a typing context without repeated variables such that $fv(M) \subseteq \{x_1, \ldots, x_n\}$. We define $\Gamma \vDash M : \phi$ as follows:

$$\Gamma \vDash M : \phi \;\stackrel{def}{\Longleftrightarrow}\; \forall P_1, \ldots, P_n \, (\forall i \;\; \vDash P_i : \pi_i) \Rightarrow (\vDash M\langle P_1/x_1\rangle \cdots \langle P_n/x_n\rangle : \phi) \,.$$

Definition 6.2. The preorder on Λ_{rc}, with associated equivalence $\simeq_{\mathscr{R}}$, is defined by

$$(M \sqsubseteq_{\mathscr{R}} N) \;\stackrel{def}{\Longleftrightarrow}\; \forall \Gamma, \phi \; (\Gamma \vDash M : \phi) \Rightarrow (\Gamma \vDash N : \phi) \,.$$

We show that this preorder contains the applicative preorder (*cf.* Definition 2.2).

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Lemma 6.3. $(M \sqsubseteq_{\mathscr{A}} N) \Rightarrow (M \sqsubseteq_{\mathscr{R}} N)$.

Proof. Assume $\Gamma = x_1 : \pi_1, \ldots, x_n : \pi_n$ and $\Gamma \models M : \phi$. We show $\Gamma \models N : \phi$ by structural induction on ϕ. If $\phi = \omega$, the statement holds by definition of $\models$. Assume $\phi = \pi \to \sigma$ and let $\models P_i : \pi_i$ for $i = 1, \ldots, n$ and $\models Q : \pi$. By assumption we have

1 $M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc}$, and
2 $\models M\langle \tilde{P}/\tilde{x}\rangle Q : \sigma$.

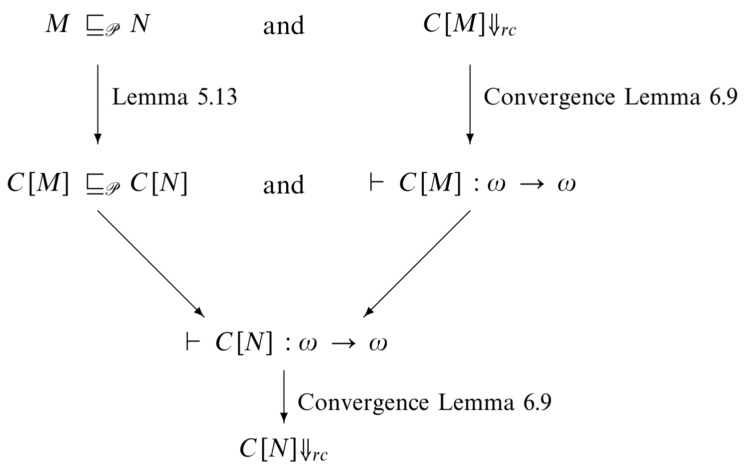
On the one hand, from (1) we have $N\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc}$, as $M \sqsubseteq_{\mathscr{A}} N$ and $[\,]\langle \tilde{P}/\tilde{x}\rangle$ is an applicative context. On the other hand, $M\langle \tilde{P}/\tilde{x}\rangle Q \sqsubseteq_{\mathscr{A}} N\langle \tilde{P}/\tilde{x}\rangle Q$ since $\sqsubseteq_{\mathscr{A}}$ is a precongruence, hence $\models N\langle \tilde{P}/\tilde{x}\rangle Q : \sigma$ by the induction hypotheses and (2). □

6.2. *Main results*

Theorem 6.4. (Full abstraction of the type semantics) For all $M, N \in \Lambda_{rc}$

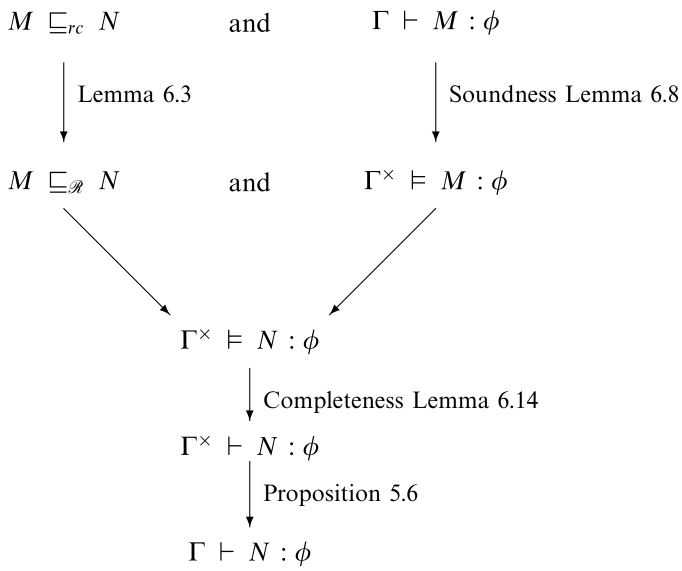
$$(M \sqsubseteq_{\mathscr{P}} N) \iff (M \sqsubseteq_{rc} N).$$

Proof. To show the $\Rightarrow$ part, called adequacy, let $M, N \in \Lambda_{rc}$ and C be a λ_r^c-context:



As for the $\Leftarrow$ part, called completeness, let $M, N \in \Lambda_{rc}$, Γ be a set of hypotheses and $\phi \in$ Ft. Then:

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press



□

Theorem 6.5. (Full abstraction of model $\mathscr{U}$) For all $M, N \in \Lambda_{rc}$

$$(M \sqsubseteq_{\mathscr{U}} N) \iff (M \sqsubseteq_{\mathscr{P}} N) \iff (M \sqsubseteq_{rc} N)\,.$$

Proof. The equivalences hold by Theorems 5.19 and 6.4. □

Remark 6.6. Without convergence testing, the model is only adequate. More precisely, if M, N range over Λ_r and contexts over λ_r-contexts, then *a fortiori* $(M \sqsubseteq_{\mathscr{U}} N) \Rightarrow (M \sqsubseteq_r N)$, but Example 2.12 and Theorem 6.5 show that the converse does not hold.

In the next three sections, we show the intermediate results (soundness, convergence, completeness). In order to establish completeness, we introduce characteristic bags and terms and justify their name in Lemma 6.12.

6.3. *Soundness*

In order to prove the soundness of $\models$ with respect to $\vdash$, we need the following technical proposition.

Proposition 6.7.

1 $(\Gamma \models M : \phi \text{ and } \phi \leqslant \sigma) \Rightarrow \Gamma \models M : \sigma\,.$
2 $(\Gamma \models M : \phi \text{ and } \Gamma \gg \Sigma) \Rightarrow \Sigma^{\times} \models M : \phi\,.$

Proof. 1. We prove the statement for closed terms, by induction on σ. For $\sigma = \omega$, it holds by definition of $\models$. If $\sigma = \psi \to \sigma'$, the type ϕ must be $\pi \to \phi'$ for some π, ϕ'. By assumption, $M{\Downarrow}_{rc}$, which proves the statement for $\sigma' = \omega$. Otherwise, given $\models P : \psi$, we show $\models MP : \sigma'$. Assuming $\psi \triangleright \phi_1, \dots, \phi_n$, the following is verified by definition of $\models$:

$$P \equiv (M_1 \mid \cdots \mid M_n \mid R) \quad \text{and} \quad \models M_i : \phi_i \quad \text{for all } i\,.$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

On the other hand, by the arrow property, we have $\psi \leqslant \pi$ and $\phi' \leqslant \sigma'$. Let $\pi \triangleright \delta_1, \ldots, \delta_m$. By the product property, there is an injection $i : [1, m] \rightarrow [1, n]$ such that $\phi_{i(j)} \leqslant \delta_j$ for all j. By induction, $\vDash M_{i(j)} : \delta_j$. Then

$$\vDash (M_{i(1)} \mid \cdots \mid M_{i(m)}) : \delta_1 \times \cdots \times \delta_m$$

implies $\vDash P : \pi$, as the types composing π different from the δ_i's are necessarily ω. By hypothesis, $\vDash MP : \phi'$. By induction, $\vDash MP : \sigma'$.

For open terms, just observe that, if $\Gamma = x_1 : \pi_1, \ldots, x : \pi_n$, the assumption writes $\vDash M\langle \tilde{P}/\tilde{x} \rangle : \phi$ for any $\tilde{P}$ such that $\vDash P_i : \pi_i$. Then $\vDash M\langle \tilde{P}/\tilde{x} \rangle : \sigma$. So $\Gamma \vDash M : \sigma$ by definition.

2. Let $\Sigma^\times(x_i) = \pi_i$. If $\Gamma(x_i) = \psi_i$, then $\pi_i \leqslant \psi_i$ by Proposition 5.6 (4). Let $\vDash P_i : \pi_i$. We have $\vDash M\langle \tilde{P}/\tilde{x} \rangle : \phi$ since Part (1) implies $\vDash P_i : \psi_i$. □

Lemma 6.8. (Soundness) For all M, Γ, and ϕ,

$$(\Gamma \vdash M : \phi) \Rightarrow (\Gamma^\times \vDash M : \phi) .$$

Proof. We use structural induction on the derivation of $\Gamma \vdash M : \phi$. We assume that $\Gamma^\times = x_1 : \pi_1, \ldots, x_k : \pi_k$ for some $k \geqslant 0$. The cases (L8) and (L9) have been already covered in proposition 6.7. We examine some other cases.

(L1) Assume $x : \phi \vdash x : \phi$. We must prove $\vDash x\langle P/x \rangle : \phi$ for any P such that $P \equiv (N \mid Q)$ and $\vDash N : \phi$. We have $x\langle P/x \rangle \rightarrow_{rc} N\langle Q/x \rangle \asymp N$. Hence $N \sqsubseteq_{\mathscr{A}} x\langle P/x \rangle$ by Lemmas 2.4 and 2.5, and $\vDash x\langle P/x \rangle : \phi$ holds by Lemma 6.3.

(L2) The induction hypothesis says that $x : \pi, \Gamma^\times \vDash N : \sigma$ (with $\phi = \pi \rightarrow \sigma$ and $M = \lambda x.N$), with $x \notin \Gamma^\times$. Let closed $P_1, \ldots, P_k, P$ be such that $\vDash P : \pi$ and $\vDash P_i : \pi_i$ for all i. By definition, we have

$$\vDash N\langle P/x \rangle \langle P_1/x_1 \rangle \cdots \langle P_k/x_k \rangle : \sigma .$$

Furthermore, since $\tilde{P}$ and P are closed and $x \neq x_i$ for any i, we have

$$(\lambda x.N)\langle \tilde{P}/\tilde{x} \rangle P \rightarrow^{\star}_{rc} (\lambda x.(N\langle \tilde{P}/\tilde{x} \rangle))P \rightarrow_{rc} N\langle \tilde{P}/\tilde{x} \rangle \langle P/x \rangle \asymp N\langle P/x \rangle \langle \tilde{P}/\tilde{x} \rangle .$$

This implies, $\vDash (\lambda x.N)\langle \tilde{P}/\tilde{x} \rangle P : \sigma$, by Lemmas 2.4, 2.5, and 6.3. Hence $\Gamma \vDash (\lambda x.N) : \pi \rightarrow \sigma$ by definition of $\vDash$.

(L3) Assume $\Sigma \vdash N : \pi \rightarrow \phi$ and $\Delta \vdash P : \pi$ (with $M = (NP)$ and $\Gamma = \Sigma, \Delta$) and let $\pi \triangleright \phi_1, \ldots, \phi_n$. By Proposition 5.7 (1), there exist some $M_1, \ldots, M_n, Q$ and $\Delta_1, \ldots, \Delta_n$ such that $P \equiv (M_1 \mid \cdots \mid M_n \mid Q)$, $\Delta_1, \ldots, \Delta_n \gg \Delta$, and $\Delta_i \vdash M_i : \phi_i$, with shorter proofs than that of $\Delta \vdash P : \pi$. By induction, we have $\Delta_i^\times \vDash M_i : \phi_i$ for $i = 1, \ldots, n$ and $\Sigma^\times \vDash N : \pi \rightarrow \phi$. Let $P_1, \ldots, P_k$ be closed bags such that $\vDash P_i : \pi_i$. Since there exists δ_i such that

$$\pi_i \sim \Sigma^\times(x_i) \times \Delta^\times(x_i) \sim \Sigma^\times(x_i) \times \Delta_1^\times(x_i) \times \cdots \times \Delta_n^\times(x_i) \times \delta_i,$$

we can find Q_i, R_j^i and S_i such that $P_i \equiv (Q_i \mid R_1^i \mid \cdots \mid R_n^i \mid S_i)$, with $\vDash Q_i : \Sigma^\times(x_i)$ and $\vDash R_j^i : \Delta_j^\times(x_i)$. Therefore, by definition of $\vDash$, the following holds:

$$\vDash N\langle \tilde{Q}/\tilde{x} \rangle : \pi \rightarrow \phi \quad \text{and} \quad \vDash N_j : \phi_j \text{ with } N_j = M_j\langle \tilde{R}_j/\tilde{x} \rangle.$$

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

By definition again, we have $\models (N_1 \mid \cdots \mid N_n \mid Q) : \pi$, and then

$$\models (N\langle\tilde{Q}/\tilde{x}\rangle)(N_1 \mid \cdots \mid N_n \mid Q) : \sigma \,.$$

By Lemma 2.7, $(N\langle\tilde{Q}/\tilde{x}\rangle)(N_1 \mid \cdots \mid N_n \mid Q) \sqsubseteq_{\mathscr{A}} N(M_1 \mid \cdots \mid M_n \mid Q)\langle\tilde{P}/\tilde{x}\rangle$. Then $\Gamma^{\times} \models (NP) : \phi$ using Lemma 6.3.

(L10) Assume $M = \mathrm{c}P$, $\phi = \sigma \to \sigma$, and $\Gamma \vdash P : \omega \to \omega$. By Proposition 5.7, there exist N, Q, Δ such that $P \equiv (N \mid Q)$, $\Delta \gg \Gamma$ and $\Delta \vdash N : \omega \to \omega$, with a shorter proof than that of $\Gamma \vdash P : \omega \to \omega$. By induction, $\Delta^{\times} \models N : \omega \to \omega$. Since $\Delta^{\times} \gg \Gamma^{\times}$, using Proposition 6.7 (2), we have $\Gamma^{\times} \models N : \omega \to \omega$. Assume $\models Q_i : \pi_i$ and $\models R : \sigma$. We must show

$$(\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle\Downarrow_{rc} \quad \text{and} \quad \models (\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle R : \sigma \,.$$

Since $\Delta^{\times} \models N : \omega \to \omega$, we have $\models N\langle\tilde{Q}/\tilde{x}\rangle : \omega \to \omega$, that is, $N\langle\tilde{Q}/\tilde{x}\rangle\Downarrow_{rc}$. Hence $\mathrm{c}(N\langle\tilde{Q}/\tilde{x}\rangle) \to_{rc}^{\star} \mathbf{I}$. Since

$$(\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle \to_{rc} (\mathrm{c}N)\langle\tilde{Q}/\tilde{x}\rangle \asymp \mathrm{c}(N\langle\tilde{Q}/\tilde{x}\rangle),$$

we have $(\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle \to_{rc}^{\star} \mathbf{I}$ (*cf.* Lemma 2.5). Then $R \sqsubseteq_{\mathscr{A}} (\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle R$, which implies $\models (\mathrm{c}P)\langle\tilde{Q}/\tilde{x}\rangle R : \sigma$ by Lemma 6.3. □

The following lemma characterizes convergence by means of the notion of typability.

Lemma 6.9. (Convergence lemma) For all closed terms M,

$$M\Downarrow_{rc} \iff \vdash (M : \omega \to \omega)\,.$$

Proof. Let $M \to_{rc}^{\star} V$, with V a value. It is easy to check that $\vdash V : \omega \to \omega$. Theorem 5.10 gives $\vdash M : \omega \to \omega$. To show the converse implication, assume $\vdash M : \omega \to \omega$. Then $\models M : \omega \to \omega$ by Lemma 6.8. That is, $M\Downarrow_{rc}$ by definition. □

An immediate consequence of Lemma 6.9 is that ω is the only type for a diverging term like Ω.

6.4. *Characteristic bags and terms*

Characteristic bags P_τ such that $\vdash P_\tau : \tau$ are defined by induction on the type τ they intend to characterize. If $\tau \in \mathrm{Ft}$, M_τ stands for P_τ. At the same time as P_τ, we construct a function T_τ, which is meant to test if its argument has type τ.

https://doi.org/10.1017/S0960129599002893 Published online by Cambridge University Press

Definition 6.10. (Characteristic bags and terms) We define P_τ and T_τ, by mutual induction on τ, as follows:

$$\begin{array}{rcl} M_\omega & = & \Omega \\ T_\omega & = & \lambda x.\mathbf{I} \end{array}$$

$$\begin{array}{rcl} M_{\pi\to\sigma} & = & \begin{cases} \lambda x.\Omega & \text{if } \sigma = \omega \\ \lambda x.(T_\pi x^\infty) M_\sigma & \text{otherwise} \end{cases} \\ T_{\pi\to\sigma} & = & \begin{cases} \lambda x.(\mathrm{c}x) & \text{if } \sigma = \omega \\ \lambda x.T_\sigma(xP_\pi) & \text{otherwise} \end{cases} \end{array}$$

$$\begin{array}{rcl} P_{\pi\times\psi} & = & (P_\pi \mid P_\psi) \\ T_\pi & = & \lambda x.(T_{\phi_1} x^\infty)\cdots(T_{\phi_n} x^\infty) \ \text{ if } \pi \rhd \phi_1,\ldots,\phi_n\,. \end{array}$$

We will comment briefly on the previous definition. For $\tau = \omega$, the choice is clear. The characteristic term of arrow type $\pi \to \sigma$ is the function (abstraction) that takes an argument of type π and gives back the characteristic term of type σ. In a typed calculus, $M_{\pi\to\sigma}$ would be $\lambda x:\pi.M_\sigma$. Instead, we use T_π to ensure that the argument of the abstraction is of type π. Observe that in the definition of $M_{\pi\to\omega}$ there is no control on the argument since $\pi \to \omega \sim \omega \to \omega$.

The definition of $T_{\pi\to\sigma}$ rests on T_σ and P_π. First suppose $\sigma \neq \omega$. By Proposition 5.7 (3), M has type $\pi \to \sigma$ if and only if MP_π has type σ. If $\sigma = \omega$, then $\pi \to \sigma \sim \omega \to \omega$, and what we have to test then is that M converges (*cf.* Lemma 6.9). Note that, in the setting of the lazy λ-calculus, a synthetic definition for the two cases just considered can be given by setting $T_{\pi\to\sigma} = \lambda x.(\mathrm{c}x)(T_\sigma(xP_\pi))$ (*cf.* Abramsky and Ong (1993) and Boudol (1990)). But that would not suit our paradigm with resources, because x occurs twice in this expression.

Characteristic bags for product types are built using parallel composition. Intuitively, the test for $\pi \times \psi$ must verify, successively, that the argument has the two types π and ψ, that is, $T_{\pi\times\psi} = \lambda x.(T_\pi x^\infty)(T_\psi x^\infty)$. But this term is not suitable for $\pi = \omega$, because it diverges when applied to a term of type $\psi \sim \omega \times \psi$. The exact definition avoids any test on type ω.

We have used infinite multiplicities in the definition of characteristic bags and terms. In fact, finite multiplicities would have been enough. The test for types in Ft uses its argument once: $T_{\pi\to\sigma}$ is linear in the abstracted variable. Moreover, the test for product types is made up of a finite number of 'simple' tests (for arrow types). Thus, infinite multiplicities can be replaced by multiplicity 1 in T_π, while the characteristic term $M_{\pi\to\sigma}$ must allow for as many fetches of its arguments as types different from ω in π. Precisely, if $\pi \rhd \phi_1,\ldots,\phi_n$, we can define $M_{\pi\to\sigma} = \lambda x.(T_\pi x^n)M_\sigma$.

Proposition 6.11.

1 $\vdash P_\tau : \tau$.
2 $\vdash T_\tau : \tau \to (\sigma \to \sigma)$.

Proof. The proof is straightforward. □

Lemma 6.12. (Characterization lemma)

1 $\Gamma \vdash P_\tau : \tau' \iff \tau \leqslant \tau'$.
2 $\Gamma \vdash T_\tau : \tau' \to (\xi \to \sigma) \iff \tau' \leqslant \tau$ and $\xi \leqslant \sigma$.

Proof. In the proof, the notation $(m\tau)$ stands for property (m) at type τ. The $\Leftarrow$ parts of (1) and (2) are consequence of Proposition 6.11 and Rule (L9). The proofs of the $\Rightarrow$ parts are by induction on the type τ and then on the sizes of type derivations. Full details are given below for the most interesting cases.

Type inferences for $M_{\pi\to\sigma}$ and $T_{\pi\times\psi}$ involve the typings of their subterms $T_\pi x^\infty$ and $T_\psi x^\infty$. In our induction, we shall make use of the fact that (2) implies the following property (3).

(3) $\Gamma \vdash T_\tau x^\infty : \xi \to \sigma \Rightarrow$
$\xi \leqslant \sigma$ and $(\tau \nsim \omega \;\Rightarrow\; \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \;\; \psi_1 \times \cdots \times \psi_n \leqslant \tau)$.

We show indeed that (2τ) implies (3τ) for all τ. If the type for $T_\tau x^\infty$ is inferred by (L3), then we have $\Gamma_1 \vdash T_\tau : \psi \to (\xi \to \sigma)$ and $\Gamma_2 \vdash x^\infty : \psi$ with $\Gamma = \Gamma_1, \Gamma_2$. By (2τ), $\psi \leqslant \tau$ and $\xi \leqslant \sigma$ hold. There are two cases to examine: if $\psi \sim \omega$, we have $\tau \sim \omega$ immediately. Otherwise, by Proposition 5.8 (4), there exist $x : \psi_1, \ldots, x : \psi_n \in \Gamma_2$ such that $\psi_1 \times \cdots \times \psi_n \leqslant \psi$. Hence $\psi_1 \times \cdots \times \psi_n \leqslant \tau$ holds by transitivity. If the proof of $\Gamma \vdash T_\tau x^\infty : \xi \to \sigma$ ends with (L8), that is $\Delta \vdash T_\tau x^\infty : \xi \to \sigma$ holds for some $\Delta \gg \Gamma$, the induction hypothesis gives $\xi \leqslant \sigma$ and

$$\tau \nsim \omega \Rightarrow \exists x : \psi'_1, \ldots, x : \psi'_m \in \Delta \;\; \psi'_1 \times \cdots \times \psi'_m \leqslant \tau\,.$$

Let $\Gamma^\times(x) = \psi_1 \times \cdots \times \psi_n$. By Proposition 5.6 (4), $\psi_1 \times \cdots \times \psi_n \leqslant \psi'_1 \times \cdots \times \psi'_m$, hence $\psi_1 \times \cdots \times \psi_n \leqslant \tau$ by transitivity. If the derivation ends with (L9), that is, $\Gamma \vdash T_\tau x^\infty : \tau'$ where $\tau' \leqslant \xi \to \sigma$, then $\tau' \sim \xi' \to \sigma'$ for some ξ' and σ' (*cf.* Remark 5.2). By induction, $\xi' \leqslant \sigma'$ and

$$\tau \nsim \omega \Rightarrow \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \;\; \psi_1 \times \cdots \times \psi_n \leqslant \tau\,.$$

If $\sigma = \omega$, then $\xi \leqslant \sigma$ holds immediately. Otherwise $\xi \leqslant \xi'$ and $\sigma' \leqslant \sigma$ hold by the arrow property, hence $\xi \leqslant \sigma$ follows by transitivity. This ends the proof of (3τ). Let us prove (1) and (2):

$\tau = \omega$:

1 The only type for Ω is ω, so $\tau \leqslant \tau' = \omega$ holds by reflexivity.
2 If the derivation of $\Gamma \vdash \lambda x.\mathbf{I} : \tau' \to (\xi \to \sigma)$ ends with (L2), that is, $x : \tau', \Gamma \vdash \mathbf{I} : \xi \to \sigma$, then $\xi \leqslant \sigma$ by Proposition 5.8 (2).

$\tau = \pi \to \omega$:

1 If the derivation ends with (L2), that is, $\tau' = \psi \to \sigma$ and $x : \psi, \Gamma \vdash \Omega : \sigma$, we have $\pi \to \omega \leqslant \omega \to \omega \leqslant \psi \to \omega$, since ω is the only type for Ω.
2 When the last rule applied in the proof of $\Gamma \vdash \lambda x.(cx) : \tau' \to (\xi \to \sigma)$ is (L2), we derive $\tau' \leqslant \omega \to \omega \sim \tau$ and $\xi \leqslant \sigma$ by Proposition 5.8 (3).

$\tau = \pi \to \phi$ and $\phi \neq \omega$:

1 We only consider the case where the derivation of $\Gamma \vdash \lambda x.(T_\pi x^\infty)M_\phi : \tau'$ ends with

(L3) followed by (L2), that is, $\tau' = \pi' \to \phi'$ and $x : \pi', \Gamma \vdash (T_\pi x^\infty)M_\phi : \phi'$ with $x \notin \Gamma$. $x : \pi', \Gamma_1 \vdash T_\pi x^\infty : \xi' \to \phi'$ and $\Gamma_2 \vdash M_\phi : \xi'$, where $\Gamma = \Gamma_1, \Gamma_2$. Since (2π) holds by induction, (3π) gives $\xi' \leqslant \phi'$ and $\pi' \leqslant \pi$. Moreover, $\phi \leqslant \xi'$ by induction, so $\phi \leqslant \phi'$ by transitivity, and hence $\pi \to \phi \leqslant \pi' \to \phi'$.

2 We will only consider the case of a derivation of $\Gamma \vdash \lambda x.T_\phi(xP_\pi) : \tau' \to (\xi \to \sigma)$, of the form

$$
\text{(L2)}\ \dfrac{\text{(L3)}\ \dfrac{\begin{array}{c}\vdots\\ \Gamma_1 \vdash T_\phi : \psi \to (\xi \to \sigma)\end{array} \quad \text{(L3)}\ \dfrac{\begin{array}{c}\vdots\\ x : \tau' \vdash x : \theta \to \psi\end{array} \quad \begin{array}{c}\vdots\\ \Gamma_2 \vdash P_\pi : \theta\end{array}}{x : \tau', \Gamma_2 \vdash xP_\pi : \psi}}{x : \tau', \Gamma_1, \Gamma_2 \vdash T_\phi(xP_\pi) : \xi \to \sigma}}{\Gamma = \Gamma_1, \Gamma_2 \vdash \lambda x.T_\phi(xP_\pi) : \tau' \to (\xi \to \sigma)}
$$

We have to show $\xi \leqslant \sigma$ and $\tau' \leqslant \tau$. We have:

— $\psi \leqslant \phi$ and $\xi \leqslant \sigma$ by the induction hypothesis on ϕ;

— $\tau' \leqslant \theta \to \psi$ by Proposition 5.8 (1); hence

— $\tau' \sim \pi' \to \phi'$ for some π', ϕ' by the product property;

— $\theta \leqslant \pi'$ and $\phi' \leqslant \psi$ by the arrow property (since $\phi \neq \omega$ implies $\psi \neq \omega$);

— $\pi \leqslant \theta$, by the induction hypothesis on π.

Putting these results together, we get $\pi \leqslant \pi'$ and $\phi' \leqslant \phi$, which implies $\tau' \sim \pi' \to \phi' \leqslant \pi \to \phi = \tau$.

$\tau = \pi \times \psi$:

1 The statement follows easily by Proposition 5.7 (1) and by induction applied to π and ψ.

2 As for the arrow case, we omit the details and only analyze the case of a derivation for $\tau = \phi_0 \times \phi_1$ with $\phi_i \neq \omega$, $\tau' = \psi_1 \times \psi_2$, and $\sigma \neq \omega$:

$$
\text{(L2)}\ \dfrac{\text{(L3)}\ \dfrac{\begin{array}{c}\vdots\\ x : \psi_1, \Gamma_1 \vdash T_{\phi_0} x^\infty : \theta \to (\xi \to \sigma)\end{array} \quad \begin{array}{c}\vdots\\ x : \psi_2, \Gamma_2 \vdash T_{\phi_1} x^\infty : \theta\end{array}}{x : \tau', \Gamma_1, \Gamma_2 \vdash (T_{\phi_0} x^\infty)(T_{\phi_1} x^\infty) : (\xi \to \sigma)}}{\Gamma = \Gamma_1, \Gamma_2 \vdash \lambda x.(T_{\phi_0} x^\infty)(T_{\phi_1} x^\infty) : \tau' \to (\xi \to \sigma)}
$$

Some consequences of this derivation are

— $\psi_1 \leqslant \phi_0$ and $\theta \leqslant \xi \to \sigma$ by $(3\phi_0)$, and hence $\theta \sim \xi' \to \sigma'$ for some ξ', σ', and $\xi \leqslant \xi'$ and $\sigma' \leqslant \sigma$;

— $\psi_2 \leqslant \phi_1$ and $\xi' \leqslant \sigma'$ by $(3\phi_1)$.

Therefore, $\tau' = \psi_1 \times \psi_2 \leqslant \phi_0 \times \phi_1 \sim \tau$ and $\xi \leqslant \xi' \leqslant \sigma' \leqslant \sigma$. □

6.5. *Completeness*

The following lemma singles out an important instance of the completeness result shown below (Lemma 6.14).

Lemma 6.13. $(\Gamma \models M : \omega \to \omega) \;\Rightarrow\; (\Gamma \vdash M : \omega \to \omega)$.

Proof. Let $\Gamma \models M : \omega \to \omega$ with $fv(M) \subseteq \{x_1, \ldots, x_n\}$ and $\Gamma(x_i) = \pi_i$ for all $1 \leqslant i \leqslant n$. By Lemma 6.11 and soundness, $\models P_{\pi_i} : \pi_i$ is verified. Then, by Lemma 6.9, $M\langle P_{\pi_1}/x_1\rangle \cdots \langle P_{\pi_n}/x_n\rangle \Downarrow_{rc}$ implies $\vdash M\langle P_{\pi_1}/x_1\rangle \cdots \langle P_{\pi_n}/x_n\rangle : \omega \to \omega$. By Proposition 5.7 (2), there exist $\psi_1, \ldots, \psi_n$ such that $\vdash P_{\pi_i} : \psi_i$ for all i and $x_1 : \psi_1, \ldots, x_n : \psi_n \vdash M : \omega \to \omega$. Characterization Lemma 6.12 gives $\pi_i \leqslant \psi_i$. The conclusion $\Gamma \vdash M : \omega \to \omega$ is obtained by (L8), since $x_1 : \psi_1, \ldots, x_n : \psi_n \gg x_1 : \pi_1, \ldots, x_n : \pi_n \gg \Gamma$. □

Lemma 6.14. (Completeness lemma) For all M, Γ, ϕ,

$$(\Gamma \models M : \phi) \;\Rightarrow\; (\Gamma \vdash M : \phi)\,.$$

Proof. The proof is by structural induction on ϕ. If $\phi = \omega$, then $\Gamma \vdash M : \omega$ by (L7). If $\phi = \pi \to \omega$, then $\Gamma \models M : \omega \to \omega$ and the statement holds by Lemma 6.13. Assume $\phi = \pi \to \sigma$ with $\sigma \neq \omega$ and $\Gamma = x_1 : \pi_1, \ldots, x_n : \pi_n$. For any $P_1, \ldots, P_n, Q$ such that $\models P_i : \pi_i$ and $\models Q : \pi$, we have

$$\begin{array}{ll} (*) & M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc} \;\text{ and} \\ (**) & \models M\langle \tilde{P}/\tilde{x}\rangle Q : \sigma\,. \end{array}$$

We claim that $(*)$ implies $\Gamma \vdash M : \omega \to \omega$, and $(**)$ implies $y : \pi, \Gamma \vdash My^\infty : \sigma$, where $y \notin \Gamma$ and $y \notin fv(M)$. The conclusion then follows from these claims by the extensionality Lemma 5.11.

We have, by the convergence Lemma 6.9 and by the soundness Lemma 6.8,

$$M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc} \;\Rightarrow\; (\vdash M\langle \tilde{P}/\tilde{x}\rangle : \omega \to \omega) \;\Rightarrow\; (\models M\langle \tilde{P}/\tilde{x}\rangle : \omega \to \omega)\,.$$

Since this holds for all $\tilde{P}$, we thus have $\Gamma \models M : \omega \to \omega$ by definition of $\models$, and $\Gamma \vdash M : \omega \to \omega$ follows by Lemma 6.13.

By Lemmas 2.5, 2.6 and 6.3, we have $M\langle \tilde{P}/\tilde{x}\rangle Q \simeq_{\mathscr{R}} (My^\infty)\langle Q/y\rangle\langle \tilde{P}/\tilde{x}\rangle$. This equivalence together with $(**)$ implies $\models (My^\infty)\langle Q/y\rangle\langle \tilde{P}/\tilde{x}\rangle : \sigma$. Hence $y : \pi, \Gamma \models My^\infty : \sigma$, and the induction hypothesis gives $y : \pi, \Gamma \vdash My^\infty : \sigma$. This completes the proof of the two claims. □

7. Conclusion

We have built up a denotational semantics for a non-confluent functional language incorporating the expression of deadlocks. A new domain equation has been proposed of the form $D = (\mathscr{M}(D) \to D)_\perp$, whose canonical solution constitutes an adequate model of the language. Moreover, completeness can be achieved by adding convergence testing facilities. Another equation of this kind, which differs in the definition of $\mathcal{M}(D)$, has been studied by the authors and presented in Lavatelli (1996). Unlike the equation presented in this paper, it allows us to interpret the whole set of arguments and not just finite ones. The corresponding canonical solution is a filter model rather than an upperset model, and turns out to be adequate. Nevertheless, the fact that the type system used to characterize this model involves a standard conjunction operator besides the product complicates the task of showing a completeness result analogous to the one presented here for the language augmented with convergence testing. The definability technique is no longer suitable for showing this: indeed, the definition of characteristic bags for conjunctive types would imply the duplication of arguments, which goes against the philosophy of the language. However, we conjecture that completeness holds for the second model as well, and we have some confidence in the technique of logical relations for proving that the canonical solutions of these two domain equations in fact induce the same preorder on terms, and hence that the second model is fully abstract too. We leave this subject for further work.

References

Abadi, M., Cardelli, L., Curien, P.-L. and Levy, J.-J. (1991) Explicit Substitutions. *Journal of Functional Programming* **1.**

Abramsky, S. (1989) *The Lazy Lambda Calculus*. In: Turner, D. (ed.) *Research Topics in Functional Programming*, Addison Wesley.

Abramsky, S. and Ong, L. (1993) Full Abstraction in the Lazy Lambda Calculus. *Information and Computation* **105** (2).

Amadio, R. and Curien, P.-L. (1998) *Domains and Lambda-Calculi*, Cambridge University Press.

Barendregt, H. P. (1984) *The Lambda Calculus*, North-Holland.

Barendregt, H. P., Coppo, M. and Dezani-Ciancaglini, M. (1983) A Filter Lambda Model and the Completeness of Type Assignment. *Journal of Symbolic Logic* **48**.

Boudol, G. (1990) A Lambda-Calculus for Parallel Functions. Rapport de Recherche **1231**, INRIA Sophia-Antipolis.

Boudol, G. (1991) Lambda-Calculi for (Strict) Parallel Functions. *Information and Computation* **108** (1).

Boudol, G. (1993) The Lambda Calculus with Multiplicities. Rapport de Recherche **2025**, INRIA Sophia-Antipolis.

Boudol, G. and Laneve, C. (1994) The Discriminating Power of Multiplicities in the λ-Calculus. *Information and Computation* **126** (1).

Boudol, G. and Laneve, C. (1995a) Termination, Deadlock and Divergence in the λ-Calculus with Multiplicities. Proc. 11th Mathematical Foundations of Programming Semantics Conference. *Electronic Notes in Computer Science*.

Boudol, G. and Laneve, C. (1995b) λ-Calculus, Multiplicities and the π-Calculus. Rapport de Recherche **2581**, INRIA Sophia-Antipolis.

Boudol, G. and Lavatelli, C. (1996) Full Abstraction for Lambda-Calculus with Resources and Convergence Testing. CAAP'96. *Springer-Verlag Lecture Notes in Computer Science* **1059**.

Coppo, M. and Dezani-Ciancaglini, M. (1980) An Extension of the Basic Functionality Theory for the λ-Calculus. *Notre Dame Journal of Formal Logic* **21**.

Coppo, M., Dezani-Ciancaglini, M., Honsell, F. and Longo, G. (1984) Extended Type Structures and Filter Lambda Models. In: Lolli, G., Longo, G. and Marcja, A. (eds.) *Logic Colloquium* **82**, Elsevier Science Publishers B.V. (North-Holland).

Coppo, M., Dezani-Ciancaglini, M. and Venneri, B. (1980)
Principal Type Schemes and λ-Calculus Semantics. In: Hindley, J. R. and Seldin, J. P. (eds.) *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press.

Coppo, M., Dezani-Ciancaglini, M. and Venneri, B. (1981) Functional Characters of Solvable Terms. *Zeit. Math. Logik Grund* **27**.

Curien, P.-L. (1998) Abstract Böhm trees. *Mathematical Structures in Computer Science* **9**.

Curien, P.-L. and Herbelin, H. (1998) Computing with abstract Böhm trees. *Proceedings of the Third Fuji International Symposium on Functional and Logic Programmming, Kyoto*.

Dezani-Ciancaglini, M., Intriglia, B. and VenturniZilli, M. (1998) Böhm's theorem for Böhm trees. *Proceedings of the Sixth Italian Conference on Theoretical Computer Science*, World Scientific.

Dezani-Ciancaglini, M., de Liguoro, U. and Piperno, A. (1994) Fully Abstract Semantics for Concurrent λ-Calculus. TACS'94. *Springer-Verlag Lecture Notes in Computer Science* **789**.

Dezani-Ciancaglini, M., de Liguoro, U. and Piperno, A. (1999a) A Filter Model for Concurrent λ-Calculus. *Siam Journal on Computing* **27** (5) 1376–1419.

Dezani-Ciancaglini, M., de Liguoro, U. and Piperno, A. (1999b) Filter Models for Conjunctive-Disjunctive λ-Calculi. *Theoretical Computer Science* **170** (1-2) 83–128.

Dezani-Ciancaglini, M. and Margaria, I. (1986) A Characterization of F-complete Type Assignments. *Theoretical Computer Science* **45**.

Hindley, J. R. (1982) The Simple Semantics for Coppo-Dezani-Sallé Types. International Symposium on Programming. *Springer-Verlag Lecture Notes in Computer Science* **137**.

Hindley, J. R. (1983) The Completeness Theorem for Typing λ-Terms. *Theoretical Computer Science* **22**.

Krivine, J.-L. (1991) *Lambda-calcul, Types et Modèles*, Masson.

Lavatelli, C. (1996) *Sémantique du Lambda Calcul avec Ressources*, Thèse de Doctorat. Université Paris VII. France.

Lévy, J.-J. (1976) An Algebraic Interpretation of the $\lambda\beta K$-Calculus; and an Application of a Labelled λ-Calculus. *Theoretical Computer Science* **2** (1).

Milner, R., Parrow, J. and Walker, D. (1989) A Calculus of Mobile Processes, Parts I and II. *Information and Computation* **100** (1).

Milner, R. (1990) Functions as Processes. *Mathematical Structures in Computer Science* **2**.

Morris, J. H. (1968) *Lambda Calculus Models of Programming Languages*, Ph. D. Thesis M.I.T.

Nielsen, M., Plotkin, G. and Winskel, G. (1981) Petri Nets, Event Structures and Domains. *Theoretical Computer Science* **13**.

Ong, C.-H. L. (1988) *The Lazy Lambda Calculus/ An Investigation into the Foundations of Functional Programming*, Ph. D. Thesis, Imperial College.

Piperno, A. (1995) Normalization and extensionality. *Proceeding of Logic in Computer Science (LICS 95)*, IEEE Computer Society Press.

Pitts, A. (1996) Relational Properties of Domains. *Information and Computation* **127**.

Plotkin, G. (1981) Post-Graduate Lecture Notes on Advanced Domain Theory, Department of Computer Science, University of Edinburgh.

Ronchi della Rocca, S. (1993) *Basic Lambda-calculus*, International Summer School in Logic for Computer Science, Université de Savoie.

Sallé, P. (1978) Une Extension de la Théorie des Types en λ-Calcul. *Springer-Verlag Lecture Notes in Computer Science* **62**.

Sangiorgi, D. (1993) *Expressing Mobility in Process Algebras: First Order and Higher Order Paradigms*, Ph. D. Thesis, Department of Computer Science, Edinburgh University.

Sangiorgi, D. (1994) The Lazy Lambda Calculus in a Concurrency Scenario. *Information and Computation* **120** (1).

Scott, D. (1982) Domains for Denotational Semantics. *Springer-Verlag Lecture Notes in Computer Science* **140**.