Linearity, Non-determinism and Solvability*

Michele Pagani
Laboratoire d'Informatique de Paris-Nord – Université Paris 13
michele.pagani@lipn.univ-paris13.fr

Simona Ronchi Della Rocca
Dipartimento di Informatica – Università di Torino
ronchi@di.unito.it

Abstract

We study the notion of solvability in the resource calculus, an extension of the λ-calculus modelling resource consumption. Since this calculus is non-deterministic, two different notions of solvability arise, one optimistic (angelical, may) and one pessimistic (demoniac, must). We give a syntactical, operational and logical characterization for the may-solvability and only a partial characterization of the must-solvability. Finally, we discuss the open problem of a complete characterization of the must-solvability.

1 Introduction

We investigate the notion of solvability in the resource calculus (Λ^r), an extension of the λ-calculus, where both the features of linearity and non-determinism are present. Λ^r is a calculus allowing to model resource consumption. Namely, the argument of a function comes as a finite multiset of resources, which in turn can be either linear or reusable. A linear resource needs to be used exactly once, while a reusable one can be called ad libitum, also zero times. Hence, the evaluation of a function applied to a multiset of resources gives rise to different possible choices, because of the different possibilities of distributing the resources among the occurrences of the formal parameter. So the calculus is not deterministic, but rather than having a purely non-deterministic notion of evaluation, the reduction introduces a formal sum of terms, representing all possible results of a non-deterministic computation. The presence of linear resources not only is a further source of non-determinism but it also introduces a notion of failure of the computation, the empty sum, distinct from non-termination: whenever the number of the available resources does not fit exactly the number of occurrences of the variable abstracted in a redex, then this latter evaluates to the empty sum. The resource calculus is a useful framework for studying the notions of linearity and non-determinism, and the relation between them.

Λ^r is an evolution of the calculus of multiplicities, this last introduced by Boudol in order to study the semantics of the lazy λ-calculus [Bou93]. Ehrhard

*Partially founded by the Italian MIUR project CONCERTO and the French ANR projet blanc CHOCO, ANR-07-BLAN-0324.

and Regnier designed the differential λ-calculus [ER03], drawing on insights gained from an analysis of some denotational models of linear logic. As the authors remarked, the differential λ-calculus seemed quite similar to Boudol's calculus of multiplicities. Indeed, this was formalized by Tranquilli, who defined the Λ^r syntax, and showed a Curry-Howard correspondence between this calculus and Ehrhard and Regnier's differential nets [Tra08]. The main differences between Boudol's calculus and Λ^r are that the former is equipped with explicit substitution and lazy operational semantics, while the latter is a true extension of the classical λ-calculus.

One way to appreciate the resource calculus is by observing the various subcalculi it contains. Intuitively, usual λ-calculus can be embedded into Λ^r translating the application MN into $M[N^!]$, where $[N^!]$ represents the multiset containing one copy of the resource N, which is reusable (see the grammar of Figure 1(a)). Forbidding linear terms but allowing non-empty finite multisets of reusable terms yields a purely non-deterministic extension of λ-calculus, which recalls de' Liguoro and Piperno's $\lambda_\oplus$-calculus [dP95] (studied also in [BEM09]). This fragment is also a variant of Vaux's algebraic λ-calculus [Vau09] without coefficients, where one moreover forbids the formation of the zero sum.

We will deal extensively with this fragment (noted here $\Lambda^\|$ and called parallel calculus), in fact all the results we have obtained for Λ^r hold also for $\Lambda^\|$. On the other side, allowing only multisets of linear terms yields the linear fragment of Λ^r, used by Ehrhard and Regnier for giving a quantitative account to λ-calculus β-reduction through Taylor expansion [ER06, ER08].

As far as the operational behaviour of Λ^r is concerned, the properties of confluence and a sort of standardization have been proved in [PT09]. In fact, confluence does not clash with non-determinism since, as written above, the outcome of a computation is a sum carrying all the possible results.

In this paper we will study the solvability property. Following the λ-calculus terminology, the word *solvable* denotes a term that can interact operationally with the environment, i.e., that can produce a given output when inserted into a context supplying it with suitable resources. According to this definition, in a computer science setting the solvable terms represent the meaningful programs.

In the λ-calculus, a closed term M is called solvable if and only if there is a sequence of arguments $\vec{N}$ such that $M\vec{N}$ reduces to the identity. λ-solvability has been completely characterized, by different points of view. Syntactically, a term is solvable if and only if it reduces to a head-normal form [Bar84], operationally, if and only if the head reduction strategy applied to it eventually stops [Bar84], logically, if and only if it can be typed in a suitable intersection type assignment system [CDCV81], denotationally, if and only if its denotation is not minimal in a suitable sensible model [Hyl76, RDRP04]. Our aim is to characterize the notion of solvability in Λ^r and $\Lambda^\|$, following the same lines.

Actually, because of the non-deterministic behaviour of the calculus, two different notions of solvability arise, one optimistic (angelical, may) and one pessimistic (demoniac, must). A closed term M is may-solvable if there is a sequence of bags $\vec{P}$ such that $M\vec{P}$ reduces to a sum in which *at least one* term of the sum is the identity, while it is must-solvable when in the final sum *all* the terms are the identity. We stress the two notions collapse and coincide with the usual notion of solvability in the fragment of Λ^r corresponding to the λ-calculus.

Our result is a characterization of the may-solvability in Λ^r (Theorem 25), and a characterization of the must-solvability in $\Lambda^{\|}$ (Theorem 36).

Theorem 25 characterizes the may-solvability from a syntactical, operational and logical point of view. An extended notion of head-normal form can be defined (called *may-outer-normal form*), such that a term is solvable if and only if it can reduce to a term of such form. From an operational point of view, we use the notion of outer-reduction strategy, defined in [PT09], where no reduction is made inside reusable resources, and we prove that in order to reach the outer-normal form we can restrict ourselves to use just reduction strategies of this kind. Also, we give a logical characterization of solvability, through a type system, assigning to terms suitable non-idempotent intersection types.

The characterization of must-solvability is a difficult problem, since it would imply to separate between the empty sum and the non-terminating terms (see discussion in Section 5). We achieve a complete characterization of must-solvability only for $\Lambda^{\|}$, the parallel fragment of Λ^r, where all resources are reusable and there is no empty sum. Inside this fragment, we characterize the must-solvability from a syntactical, operational and logical point of view, through a notion of *must-outer-normal form*.

All these characterizations are conservative with respect to the λ-calculus.

The type assignment system characterizing may-solvability is strongly related to the relational semantics of linear logic. It can be seen, basically, as an extension to Λ^r of the type system introduced by de Carvalho in the restricted case of λ-calculus [dC09]. On the other side, the type assignment system characterizing must-solvability is based on a type theory supplying a logical description of the D_∞ model of Scott [Sco76, PRDR04].

The characterization of the may-solvability in Λ^r has been already presented, in a preliminary version, at FOSSACS 2010 [PRDR10].

The paper is organized as follows. Section 2 contains a syntactical description of the resource calculus. Section 3 is dedicated to the definition of may- and must-solvability and of outer-normal form. In Section 4, the complete characterization of may-solvability is given. In Section 5, the problem of the characterization of the must-solvability is discussed, and its characterization is given, inside the non-deterministic fragment $\Lambda^{\|}$.

2 Resource λ-calculus

The syntax of Λ^r. Basically, we have three syntactical sorts: terms, that are in functional position, bags, that are in argument position and represent multisets of resources, and finite formal sums, that represent the possible results of a computation. Precisely, Figure 1(a) gives the grammar for generating the set Λ^r of **terms** and the set Λ^b of **bags** (which are in fact finite multisets of **resources** $\Lambda^{(!)}$) together with their typical metavariables. A resource can be linear (it must be used exactly once) or not (it can be used ad libitum, also zero times), in the last case it is written with a ! apex. Bags are multisets presented in multiplicative notation, so that $P\cdot Q$ is the multiset union, and $1 = [\,]$ is the empty bag: that means, $P\cdot 1 = P$ and $P\cdot Q = Q\cdot P$. It must be noted though that we will never omit the dot $\cdot$, to avoid confusion with application.

Sums are multisets in additive notation, with 0 referring to the empty multiset, so that: $\mathbb{M}+0 = \mathbb{M}$ and $\mathbb{M}+\mathbb{N} = \mathbb{N}+\mathbb{M}$. We use two different notations for multisets in order to underline the different role of bags and sums. The former

Λ^r:	M, N, L	$::= x \mid \lambda x.M \mid MP$	terms
$\Lambda^{(!)}$:	$M^{(!)}, N^{(!)}$	$::= M \mid M^!$	resources
Λ^b:	P, Q, R	$::= 1 \mid [M^{(!)}] \mid P{\cdot}Q$	bags
$\Lambda^{(b)}$:	A, B	$::= M \mid P$	expressions
$\mathtt{Nat}\langle\Lambda^r\rangle$:	$\mathbb{M}, \mathbb{N}, \mathbb{L}$	$::= 0 \mid M \mid \mathbb{M} + \mathbb{N}$	sums of terms
$\mathtt{Nat}\langle\Lambda^b\rangle$:	$\mathbb{P}, \mathbb{Q}, \mathbb{R}$	$::= 0 \mid P \mid \mathbb{P} + \mathbb{Q}$	sums of bags
$\mathbb{A}, \mathbb{B} \in \mathtt{Nat}\langle\Lambda^{(b)}\rangle := \mathtt{Nat}\langle\Lambda^r\rangle \cup \mathtt{Nat}\langle\Lambda^b\rangle$			sums of expressions

(a) Grammar of terms, bags, sums, expressions.

$$\lambda x.(\textstyle\sum_i M_i) := \sum_i \lambda x.M_i \qquad (\sum_i M_i)P := \sum_i M_i P \qquad M(\sum_i P_i) := \sum_i MP_i$$

$$[(\textstyle\sum_i M_i)]{\cdot}P := \sum_i [M_i]{\cdot}P \qquad [(\sum_i M_i)^!]{\cdot}P := [M_1^!, \dots, M_k^!]{\cdot}P$$

(b) Notation on $\mathtt{Nat}\langle\Lambda^{(b)}\rangle$.

Figure 1: Syntax of the resource calculus.

are multisets in argument position, while the latter are in functional position: the two define notions of parallel composition behaving quite differently, as we will see in a while.

An **expression** (whose set is denoted by $\Lambda^{(b)}$) is either a term or a bag. Though in practice only sums of terms are needed, for the sake of the proofs we also introduce sums of bags and of expressions. The symbol $\mathtt{Nat}$ denotes the set of natural numbers, so that $\mathtt{Nat}\langle\Lambda^r\rangle$ (resp. $\mathtt{Nat}\langle\Lambda^b\rangle$) can be seen as the set of finite formal sums of terms (resp. bags). In particular, for $n \in \mathtt{Nat}$ and A an expression, the writing nA denotes the sum $\underbrace{A + \cdots + A}_{n\text{times}}$. In writing $\mathtt{Nat}\langle\Lambda^{(b)}\rangle$ we are abusing the notation, as it does not denote the $\mathtt{Nat}$-module generated over $\Lambda^{(b)} = \Lambda^r \cup \Lambda^b$ but rather the union of the two $\mathtt{Nat}$-modules. This amounts to say that sums may be taken only in the same sort.

The grammar for terms and bags does not include sums in any point, so that in a sense they may arise only as a top level constructor. However, as an inductive notation (and *not* in the actual syntax) we extend all the constructors to sums as shown in Figure 1(b). In fact, all constructors but the $(\cdot)^!$ are, as expected, linear in the algebraic sense, i.e. they commute with sums.[1] In particular, we have that 0 is always absorbing but for the $(\cdot)^!$ constructor, in which case we have $[0^!] = 1$. Notice the similarity between the equations $[0^!] = 1$ and $[(M + N)^!] = [M^!]{\cdot}[N^!]$ and, respectively, $\mathrm{e}^0 = 1$ and $\mathrm{e}^{x+y} = \mathrm{e}^x \cdot \mathrm{e}^y$: this is far from a coincidence, as the relation between Taylor expansion and linear logic semantics shows [ER08]. We refer to [Tra08, Tra09] for the mathematical intuitions underlying the resource calculus.

We adopt the usual λ-calculus conventions as in [Bar84]. Also, we use the

[1] However, the constructors do not commute with bag product, i.e. $M(P{\cdot}Q) \neq MP + MQ$. This is a first motivation for the different multiset notation between sums and bags.

$$
y\langle N/x\rangle := \begin{cases} N & \text{if } y = x, \\ 0 & \text{otherwise,} \end{cases} \qquad
\begin{aligned} (\lambda y.M)\langle N/x\rangle &:= \lambda y.(M\langle N/x\rangle), \\ (MP)\langle N/x\rangle &:= M\langle N/x\rangle P + M(P\langle N/x\rangle), \end{aligned}
$$

$$
[M]\langle N/x\rangle := [M\langle N/x\rangle], \qquad 1\langle N/x\rangle := 0,
$$

$$
[M^!]\langle N/x\rangle := [M\langle N/x\rangle, M^!], \qquad (P\cdot R)\langle N/x\rangle := P\langle N/x\rangle\cdot R + P\cdot R\langle N/x\rangle.
$$

Figure 2: Linear substitution. In the abstraction case we suppose $y \notin \mathrm{FV}(N) \cup \{x\}$.

following notation for terms useful to build examples:

$$
\mathbf{I} := \lambda x.x\,, \qquad \mathbf{F} := \lambda xy.y\,, \qquad \mathbf{\Delta} := \lambda x.x[x^!]\,, \qquad \mathbf{\Omega} := \mathbf{\Delta}[\mathbf{\Delta}^!]\,.
$$

There is no technical difficulty in defining α-equivalence and the set $\mathrm{FV}(\mathbb{A})$ of free variables as in ordinary λ-calculus. The symbol $=$ will denote the α-equivalence between expressions.

The pair reusable/linear has a counterpart in the following two different notions of substitutions: their definition, hence that of reduction, heavily uses the notation of Figure 1(b).

Definition 1 (Substitutions)**.** *We define the following substitution operations.*

1. *$A\{N/x\}$ is the usual λ-calculus (i.e. capture free) substitution of N for x. It is extended to sums as in $\mathbb{A}\{\mathbb{N}/x\}$ by linearity in $\mathbb{A}$*[2] *and using the notations of Figure 1(b) for $\mathbb{N}$. The form $A\{x + N/x\}$ is called **partial substitution**.*

2. *$A\langle N/x\rangle$ is the **linear substitution** defined inductively in Figure 2. It is extended to $\mathbb{A}\langle \mathbb{N}/x\rangle$ by bilinearity in both $\mathbb{A}$ and $\mathbb{N}$.*

Roughly speaking, the linear substitution corresponds to the replacement of the resource to exactly one *linear* occurrence of the variable. In the presence of multiple occurrences, all the possible choices are made, and the result is the sum of them. For example $(y[x][x])\langle N/x\rangle = y[N][x] + y[x][N]$. In the case there are no free linear occurrences, then linear substitution returns 0, morally an error message. For example $(\lambda y.y)\langle N/x\rangle = \lambda y.(y\langle N/x\rangle) = \lambda y.0 = 0$. Finally, in case of reusable occurrences of the variable, linear substitution acts on a linear copy of the variable, e.g. $[x^!]\langle N/x\rangle = [N, x^!]$. Indeed linear substitution bears resemblance to differentiation, as it is in Ehrhard and Regnier's differential λ-calculus [ER03], so deriving a constant function (as is $\lambda y.y$ with respect to the parameter x) returns 0 and deriving a product of functions (as is $y[x][x]$ with respect to x) gives a sum. Also the rule $[M^!]\langle N/x\rangle = [M\langle N/x\rangle, M^!]$ resembles the chain rule. The following are examples with sums

$$
\begin{aligned}
(x[x^!])\langle (M{+}N)/x\rangle &= (x[x^!])\langle M/x\rangle + (x[x^!])\langle N/x\rangle \\
&= M[x^!] + x[M, x^!] + N[x^!] + x[N, x^!], \\
(x[x^!])\,\{(M+N)/x\} &= (M+N)[(M+N)^!] = M[M^!, N^!] + N[M^!, N^!].
\end{aligned}
$$

Substitutions commute as stated in the following.

[2] $F(A)$ (resp. $F(A,B)$) is extended by linearity (resp. bilinearity) by setting $F(\sum_i A_i) = \sum_i F(A_i)$ (resp. $F(\sum_i A_i, \sum_j B_j) = \sum_{i,j} F(A_i, B_j)$).

Lemma 2 ([ER03, Tra08, Tra09])**.** *For $\mathbb{A}$ a sum of expressions, $\mathbb{M}, \mathbb{N}$ sums of terms and x, y variables such that $y \notin \mathrm{FV}(\mathbb{M}) \cup \mathrm{FV}(\mathbb{N})$, we have*

$$\begin{aligned}\big(\mathbb{A}\langle\mathbb{N}/y\rangle\big)\langle\mathbb{M}/x\rangle &= \big(\mathbb{A}\langle\mathbb{M}/x\rangle\big)\langle\mathbb{N}/y\rangle + \mathbb{A}\langle\mathbb{N}\langle\mathbb{M}/x\rangle/y\rangle \\ \big(\mathbb{A}\left\{y+\mathbb{N}/y\right\}\big)\langle\mathbb{M}/x\rangle &= \big(\mathbb{A}\langle\mathbb{M}/x\rangle\big)\left\{y+\mathbb{N}/y\right\} + \mathbb{A}\langle\mathbb{N}\langle\mathbb{M}/x\rangle/y\rangle\left\{y+\mathbb{N}/y\right\}.\end{aligned}$$

In particular, if $x \notin \mathrm{FV}(\mathbb{N})$ then the second term of both sums is 0 and the two substitutions commute. Furthermore if $x \notin \mathrm{FV}(\mathbb{M}) \cup \mathrm{FV}(\mathbb{N})$,

$$(\mathbb{A}\left\{x+\mathbb{M}/x\right\})\left\{x+\mathbb{N}/x\right\} = \mathbb{A}\left\{x+\mathbb{M}+\mathbb{N}/x\right\} = (\mathbb{A}\left\{x+\mathbb{N}/x\right\})\left\{x+\mathbb{M}/x\right\}.$$

The reductions of Λ^r. A (monic) context $C(\cdot)$ is a term that uses a distinguished free variable called its **hole** exactly once. Formally, the set of **simple contexts** is given by the following grammar

$$\Lambda_{(\cdot)}: \qquad C(\cdot), D(\cdot) ::= (\cdot) \mid \lambda x.C(\cdot) \mid C(\cdot)P \mid M[C(\cdot)]\cdot P \mid M[(C(\cdot))^!]\cdot P$$

A **context** $\mathbb{C}(\cdot)$ is a simple context in $\Lambda_{(\cdot)}$ summed to any sum in $\mathtt{Nat}\langle\Lambda^r\rangle$. The expression $\mathbb{C}(M)$ denotes the result of blindly replacing M to the hole (allowing variable capture) in $\mathbb{C}(\cdot)$. We generalize to sums applying the notations of Figure 1(b). For example $C(\cdot) := \lambda x.y[(\cdot)^!]$ and $D(\cdot) := \lambda x.y[(\cdot)]$ are simple contexts. If $\mathbb{M} = x+y$, then $C(\mathbb{M}) = \lambda x.y[x^!, y^!]$ and $D(\mathbb{M}) = \lambda x.y[x] + \lambda x.y[y]$.

A relation $\tilde{r}$ in $\Lambda^r \times \mathtt{Nat}\langle\Lambda^r\rangle$ is extended to one in $\mathtt{Nat}\langle\Lambda^r\rangle \times \mathtt{Nat}\langle\Lambda^r\rangle$ by **context closure**[3] by setting: $\mathbb{M}\ \tilde{r}\ \mathbb{N}$ iff $\exists\mathbb{C}(\cdot)$ and $M'\ \tilde{r}\ \mathbb{N}'$ such that $\mathbb{M} = \mathbb{C}(M'), \mathbb{N} = \mathbb{C}(\mathbb{N}')$.

A context is **linear** if its hole is not under the scope of a $(\,)^!$ operator. Linear contexts can be defined inductively omitting the case $M[(C(\cdot))^!]\cdot P$ in the $\Lambda_{(\cdot)}$ definition. An **outer-context** is a context having the hole not in a bag. Outer-contexts can be defined inductively omitting the cases $M[C(\cdot)]\cdot P$ and $M[(C(\cdot)^!)]\cdot P$ in the $\Lambda_{(\cdot)}$ definition. Notice that the composition $\mathbb{C}(\mathbb{D}(\cdot))$ of two outer- (resp. linear) contexts $\mathbb{C}(\cdot)$, $\mathbb{D}(\cdot)$ is an outer- (resp. linear) context.

We define two kinds of reduction rule, baby-step and giant-step reduction, the former being a decomposition of the latter. Both are meaningful: baby-step is more atomic, performing one substitution at a time, while the giant-step is closer to λ-calculus β-reduction, wholly consuming its redex in one shot.

Definition 3 ([Tra08, Tra09])**.** *The **baby-step** reduction $\xrightarrow{\mathtt{b}}$ is defined by the context closure of the following relation (assuming x not free in N):*

$$(\lambda x.M)1 \xrightarrow{\mathtt{b}} M\left\{0/x\right\} \qquad (\lambda x.M)[N]\cdot P \xrightarrow{\mathtt{b}} (\lambda x.M\langle N/x\rangle)P$$

$$(\lambda x.M)[N^!]\cdot P \xrightarrow{\mathtt{b}} (\lambda x.M\left\{N+x/x\right\})P$$

*The **giant-step** reduction $\xrightarrow{\mathtt{g}}$ is defined by the context closure of the following relation, for $\ell, n \geq 0$ (assuming x not free in the L_i's and N_i's):*

$$(\lambda x.M)[L_1,\ldots,L_\ell,N_1^!,\ldots,N_n^!] \xrightarrow{\mathtt{g}} M\langle L_1/x\rangle\ldots\langle L_\ell/x\rangle\left\{N_1+\cdots+N_n/x\right\}.$$

In the case $n=0$, the reduct is $M\langle L_1/x\rangle\ldots\langle L_\ell/x\rangle\left\{0/x\right\}$. For any reduction $\xrightarrow{\mathtt{x}}$, we denote by $\xrightarrow{\mathtt{x}+}$ and $\xrightarrow{\mathtt{x}*}$ its transitive and reflexive-transitive closure respectively.

[3] In [Tra08, Tra09] bag contexts are defined too, so that context closure extends a relation to $\mathtt{Nat}\langle\Lambda^{(b)}\rangle \times \mathtt{Nat}\langle\Lambda^{(b)}\rangle$. In fact we prefer to introduce the term contexts only, making clear that the set $\mathtt{Nat}\langle\Lambda^r\rangle$ is the actual protagonist of the calculus. However, our choice is a matter of taste, affecting no main property of the calculus.

$$
\begin{tikzcd}
(\lambda x.x[\mathbf{\Delta}^!])[\mathbf{\Delta}^!, \mathbf{I}] \arrow[r, "\mathtt{b}"] \arrow[ddrr, bend right, "\mathtt{g}"'] & (\lambda x.(\mathbf{\Delta} + x)[\mathbf{\Delta}^!])[\mathbf{I}] \arrow[r, "\mathtt{b}"] & (\lambda x.x[\mathbf{\Delta}^!])[\mathbf{I}] \arrow[d, "\mathtt{b}"] & \\
 & & (\lambda x.\mathbf{I}[\mathbf{\Delta}^!])1 \arrow[d, "\mathtt{b}"] & \\
 & & \mathbf{I}[\mathbf{\Delta}^!] \arrow[r, "\mathtt{b}"] \arrow[dr, bend right, "\mathtt{g}"'] & (\lambda x.\mathbf{\Delta})1 \arrow[d, "\mathtt{b}"] \\
 & & & \mathbf{\Delta}
\end{tikzcd}
$$

Figure 3: An example of baby-step and giant-step reductions. We use the notation of Figure 1(b): after the first $\mathtt{b}$-step the term $(\lambda x.(\mathbf{\Delta} + x)[\mathbf{\Delta}^!])[\mathbf{I}]$ stands for $(\lambda x.\mathbf{\Delta}[\mathbf{\Delta}^!])[\mathbf{I}] + (\lambda x.x[\mathbf{\Delta}^!])[\mathbf{I}]$, and the term after the following step is equal to $0 + (\lambda x.x[\mathbf{\Delta}^!])[\mathbf{I}]$. In fact 0 is the neutral element of the sum and $(\lambda x.\mathbf{\Delta}[\mathbf{\Delta}^!])[\mathbf{I}] \xrightarrow{\mathtt{b}} 0$.

Notice that giant-step reduction is defined independently of the ordering of the linear substitutions, as shown by the substitution commutations stated in Lemma 2. Baby-step and giant-step reductions are clearly related to each other.

Proposition 4 ([Tra08, Tra09])**.** *We have $\xrightarrow{\mathtt{g}} \subset \xrightarrow{\mathtt{b}*} \subset \xrightarrow{\mathtt{g}*}\xleftarrow{\mathtt{g}*}$, where the last denotes the relational composition between $\xrightarrow{\mathtt{g}*}$ and its inverse $\xleftarrow{\mathtt{g}*}$.*

Figure 3 shows an example of baby-step and giant-step reduction sequences. The reader can check, in this example, that the two reductions are related to each other as stated in Proposition 4. By the way, let us mention that although giving the same normal forms, baby-step and giant-step reductions might have different properties: for example, the starting term in the Figure 3 is strongly normalizing for giant-step but only weakly normalizing for baby-step reduction (an infinite reduction sequence can be obtained by firing the $\mathbf{\Delta}[\mathbf{\Delta}^!]$ redex in the first term of the sum $(\lambda x.(\mathbf{\Delta} + x)[\mathbf{\Delta}^!])[\mathbf{I}]$).

The fragment $\Lambda^{\|}$. We will consider a notable fragment of Λ^r, denoted by $\Lambda^{\|}$ and called the **parallel fragment**. It corresponds to a purely non-deterministic extension of λ-calculus which recalls de' Liguoro and Piperno's $\lambda_{\oplus}$-calculus [dP95]. Seeing the product of bags as a convolution product and taking into account that $[(M+N)^!] = [M^!]\cdot[N^!]$, the fragment $\Lambda^{\|}$ is also a fragment of Vaux's algebraic λ-calculus [Vau09], without coefficients nor zero sums.

$\Lambda^{\|}$ is defined permitting only non-empty bags with reusable resources and non-zero sums. Formally, this is obtained by replacing the grammar defining Λ^b (Figure 1(a)) by the following:

$$\Lambda^{b\|}: \quad P, Q, R ::= [M^!] \mid P\cdot Q$$

and by considering only non-zero sums, i.e. the sets $\mathtt{Nat}^+\langle\Lambda^{\|}\rangle$ and $\mathtt{Nat}^+\langle\Lambda^{b\|}\rangle$ of formal sums with coefficients in the set $\mathtt{Nat}^+$ of non-zero natural numbers.

Notice $\Lambda^{\|}$ is not closed under baby step reduction, since the latter can create empty bags. We therefore slightly adapt such a reduction as follows.

Definition 5. *The **middle-step** reduction $\xrightarrow{\mathtt{m}}$ is defined by the context closure of the following relation (assuming x not free in N):*

$$(\lambda x.M)[N^!] \xrightarrow{\mathtt{m}} M\{N/x\}, \qquad (\lambda x.M)[N^!]\cdot[L^!]\cdot P \xrightarrow{\mathtt{m}} (\lambda x.M\{N+x/x\})[L^!]\cdot P.$$

Clearly middle-step reduction does not alter Λ^r operational semantics.

Proposition 6. *We have* $\xrightarrow{\mathtt{g}} \subset \xrightarrow{\mathtt{m}*} \subset \xrightarrow{\mathtt{b}*}$ *in* Λ^r.

The outer reduction. The notion of head-reduction is extended to such a setting as follows.

Definition 7 ([PT09])**.** *Let* $\epsilon \in \{\mathtt{b}, \mathtt{g}, \mathtt{m}\}$. *The* ***outer*** $\boldsymbol{\epsilon}$***-reduction*** $\xrightarrow{\mathtt{o}\epsilon}$ *is the closure to* linear *contexts of the* ϵ *steps given in Definition 3 and 5.*

Notice that an outer redex (i.e. a redex for $\xrightarrow{\mathtt{o}\epsilon}$) is a redex not under the scope of a $(\cdot)^!$ constructor. For the terms in $\Lambda^{\|}$ (so in particular for the terms corresponding to the λ-terms) the outer redexes are exactly the head-redexes. However, in terms with linear resources it is possible to have outer redexes also in argument position, e.g. in $x[\mathbf{I}[y]]$.

3 May and Must Solvability

A λ-term is solvable whenever there is a outer-context reducing it to the identity [Bar84]. In the resource calculus, terms appear in formal sums, where repetitions do matter, so various notions of solvability arise, depending on the number of times one gets the identity. We deal extensively with the two different notions of solvability, related to a may and must operational semantics, respectively.

Definition 8. *A simple term M is* ***may-solvable*** *whenever there are a simple outer-context $C(\cdot)$ and a sum of terms $\mathbb{N}$, possibly 0, such that $C(M) \xrightarrow{\mathtt{g}*} \mathbf{I} + \mathbb{N}$; M is* ***must-solvable*** *whenever there are a simple outer-context $C(\cdot)$ and a number $n \neq 0$ such that $C(M) \xrightarrow{\mathtt{g}*} n\mathbf{I}$. We will say M is* ***may-solvable*** *(resp.* ***must-solvable****)* ***in*** $\Lambda^{\|}$ *in case the context $C(\cdot)$ is in $\Lambda^{\|}$.*

The above definition considers the giant-step reduction, however, one can replace it with the baby-step or middle-step reductions, obtaining an equivalent notion of solvability, as easily argued from Proposition 4 and 6. The restriction to *simple* and *outer*-contexts is crucial: there are general contexts reducing constantly to $\mathbf{I}$, disregarding the term they are applied to. Let $\mathbb{C}(\cdot)$ be either the outer- but non-simple context $\mathbf{I} + (\cdot)[\mathbf{I}1]$ or the simple but non-outer-context $(\lambda x.\mathbf{I})[(\cdot)^!]$: we have $\mathbb{C}(M) \xrightarrow{\mathtt{g}} \mathbf{I}$ for every term M.

Clearly, must-solvability implies the may-solvability but not vice versa. Consider, for example, the following reductions, where $\mathbf{I}$ and $\boldsymbol{\Omega}$ have been defined in the previous section:

$$\mathbf{I}[\mathbf{I}, \boldsymbol{\Omega}] \xrightarrow{\mathtt{g}} 0 \tag{1}$$

$$\mathbf{I}[\mathbf{I}^!, \boldsymbol{\Omega}] \xrightarrow{\mathtt{g}} \boldsymbol{\Omega} \tag{2}$$

$$\mathbf{I}[\mathbf{I}, \boldsymbol{\Omega}^!] \xrightarrow{\mathtt{g}} \mathbf{I} \tag{3}$$

$$\mathbf{I}[\mathbf{I}^!, \boldsymbol{\Omega}^!] \xrightarrow{\mathtt{g}} \mathbf{I} + \boldsymbol{\Omega} \tag{4}$$

We can reasonably guess that $\boldsymbol{\Omega}$ is neither may-solvable nor must-solvable (this will be proved in what follows), as well as any term reducing to 0. Hence, (1) and (2) give examples of unsolvable terms, (3) is both may- and must-solvable and

(4) is a may-solvable but not must-solvable term. One major outcome of this paper is the operational characterization of solvability by means of the following notions of outer normalizability.

Definition 9. *An expression is an* ***outer-normal form****,* onf *for short, if it has no redex but under the scope of a* $()^!$. *The set* onf *can be defined inductively as follows.*

$$\frac{M \in \mathrm{onf}}{\lambda x.M \in \mathrm{onf}} \qquad \frac{P_i \in \mathrm{onf},\ \textit{for } i \leq p}{xP_1 \dots P_p \in \mathrm{onf}}\ p \geq 0$$

$$\frac{}{1 \in \mathrm{onf}} \qquad \frac{M \in \mathrm{onf}}{[M] \in \mathrm{onf}} \qquad \frac{}{[M^!] \in \mathrm{onf}} \qquad \frac{P \in \mathrm{onf} \quad Q \in \mathrm{onf}}{P \cdot Q \in \mathrm{onf}}$$

A sum $\mathbb{M}$ *of terms is a* ***may-outer-normal form****,* monf *for short, whenever it contains a term in outer-normal form;* $\mathbb{M}$ *is a* ***must-outer-normal form****,* Monf *for short, whenever it is a non-empty sum of outer-normal forms. A term* M *is* ***may-/must-outer normalizable*** *iff it is reducible to a may-/must-outer-normal form.*

Notice that the Monf are the normal forms with respect to the outer reductions defined in Definition 7. Let us restrict ourselves to consider resource terms which correspond to λ-terms, according to the standard correspondence recalled in the introduction. Then the two definitions of may and must solvability coincide, and they become the standard λ-solvability. Moreover the two notions of may and must-outer-normal form collapse, and they become the classical definition of head-normal form; indeed the only possible outer redex in such terms is the head-redex.

The term $\lambda x.y1[x^!, \mathbf{\Omega}^!]$ is a onf, and it is must-solvable (hence may-solvable) via $(\lambda y.(\cdot))[\mathbf{F}^!][\mathbf{I}]$: indeed, $(\lambda y.(\lambda x.y1[x^!, \mathbf{\Omega}^!]))[\mathbf{F}^!][\mathbf{I}] \xrightarrow{\mathsf{g}} (\lambda x.\mathbf{F}1[x^!, \mathbf{\Omega}^!])[\mathbf{I}] \xrightarrow{\mathsf{g}} (\lambda x.\mathbf{I}[x^!, \mathbf{\Omega}^!])[\mathbf{I}] \xrightarrow{\mathsf{g}} \mathbf{I}[\mathbf{I}] + (\lambda x.\mathbf{\Omega})[\mathbf{I}] \xrightarrow{\mathsf{g}*} \mathbf{I} + 0 = \mathbf{I}$. The terms $\mathbf{F}[\mathbf{I}^!, \mathbf{\Omega}^!]$, $\mathbf{I}[\mathbf{I}^!, \mathbf{\Omega}^!]$ are not onf but the first is must-outer normalizable, the second is only may-outer normalizable (the former reducing to $\mathbf{I}$, the latter to $\mathbf{I} + \mathbf{\Omega}$); indeed, the first is also must-solvable, the second only may-solvable. The terms $\mathbf{F}[x]$ or $\mathbf{I}[\mathbf{\Omega}^!]$ are not may/must-outer normalizable: they reduce to 0 and $\mathbf{\Omega}$, respectively.

The main result of this section is to prove that may-outer and must-outer normalizability entails may-solvability and must-solvability respectively (Theorem 13). Section 4 will prove the converse of this implication with respect to the may-solvability, while Section 5 discusses the case of the must-solvability, where the implication *must-solvable* ⇒ *must-outer normalizable* does not hold in general but in $\Lambda^{\parallel}$.

In contrast to the λ-calculus, where having a head-normal form straightforwardly implies being solvable (see e.g. [Bar84]), Theorem 13 is not immediate, for two reasons. In the λ-calculus one can replace the head-variable of a onf with a term erasing all its arguments and returning the identity. In the resource calculus this is not possible, because of the linear resources, that cannot be erased but must be used. A further difficulty arises in the must case. In fact, given a term $\mathbb{M}$ in must-outer normal form, one should define a simple outer-context $C(\cdot)$ such that $C(M) \xrightarrow{\mathsf{g}*} n_M\mathbf{I}$ for *all* terms M in $\mathbb{M}$. This is not at all obvious since the terms in $\mathbb{M}$ can be quite different each other. Such a context is built in the proof of Lemma 12 and it is based on the notion of *resource* underlined by this calculus. In Definition 10 we give a notion of query of resources of a onf, then, roughly speaking, what Lemma 12 shows is that one can build for any term M in the sum $\mathbb{M}$ a context $C_M(\cdot)$ such that $C_M(M') \xrightarrow{\mathtt{g}*} n_{M'}\mathbf{I}$ whenever M' is a term querying the same number of resources as M and $C_M(M') \xrightarrow{\mathtt{g}*} 0$ whenever M' queries more resources than M. Then Theorem 13 is achieved taking $C_M(\cdot)$ for M a term of the sum $\mathbb{M}$ querying the minimum amount of resources among the simple terms in $\mathbb{M}$.

Definition 10. *The* query of resources *of a* onf A *is the number* $\mathrm{r}(A)$ *of variable occurrences (both bound and free) which are not under the scope of a* $(\,)^!$ *constructor, i.e.:*

$$\mathrm{r}(\lambda x.M) := \mathrm{r}(M) \qquad \mathrm{r}(xP_1 \dots P_p) := 1 + \sum_{i=1}^{p} \mathrm{r}(P_i)$$

$$\mathrm{r}([M]) := \mathrm{r}(M) \qquad \mathrm{r}([M^!]) := 0 \qquad \mathrm{r}(1) := 0 \qquad \mathrm{r}(P \cdot Q) := \mathrm{r}(P) + \mathrm{r}(Q)$$

Clearly for any onf $y\vec{P}$ and term N, we have $\mathrm{r}(y\vec{P}) = \mathrm{r}\big(y\vec{P}[N^!]\big)$.

Lemma 11. *For every* $n \in \mathtt{Nat}$*, let* $\mathbf{X}_n := \lambda x_1 \dots x_{n+1}.x_{n+1}[x_1^!, \dots, x_n^!]$*. Let* A *be a* onf*,* x *be a variable free in* A*, and* n *be any number greater than the size of* A*. Then* $A\{\mathbf{X}_n/x\}$ $\mathtt{g}$*-reduces to a* onf *having the same query of resources as* A *and no free occurrence of* x*.*

Proof. By induction on the structure of A. The only interesting case is when $A = xP_1 \dots P_p$, where by hypothesis $p \leq n$. By induction each $P_i\{\mathbf{X}_n/x\}$ $\mathtt{g}$-reduces to a bag $\hat{P}_i$ in onf with query of resources equal to that of P_i, so

$$\begin{aligned} M\{\mathbf{X}_n/x\} &= \mathbf{X}_n\big(P_1\{\mathbf{X}_n/x\}\big) \dots \big(P_p\{\mathbf{X}_n/x\}\big) \\ &\xrightarrow{\mathtt{g}*} \lambda x_{n+1-p} \dots x_{n+1}.x_{n+1}\hat{P}_1 \cdot \ \dots \ \cdot \hat{P}_p \cdot [x_{p+1}^!, \dots, x_n^!] \end{aligned}$$

The last term is a onf with query of resources equal to $1 + \sum_{i=1}^{p} \mathrm{r}(\hat{P}_i) = 1 + \sum_{i=1}^{p} \mathrm{r}(P_i) = \mathrm{r}(M)$. ☐

Lemma 12. *Let* $\mathbb{M}$ *be a must-outer-normal form. There is a simple outer-context* $C(\cdot)$ *such that* $C(\mathbb{M}) \xrightarrow{\mathtt{g}*} n\mathbf{I}$*, for* $n \in \mathtt{Nat}^{>0}$*. If moreover* $\mathbb{M} \in \Lambda^{\|}$*, then also* $C(\cdot) \in \Lambda^{\|}$*.*

Proof. In fact we prove a stronger statement:

($*$) let $\mathbb{M} = \sum_{i=1}^{m} M_i$ be a must-outer-normal form, s.t. $\mathrm{r}(M_1) = \min_{i \leq m} \mathrm{r}(M_i)$, then there is a simple outer-context $C(\cdot)$ and numbers $\{n_i\}_{i \leq m}$ such that for every $i \leq m$,

1. $C(M_i) \xrightarrow{\mathtt{g}*} n_i\mathbf{I}$,
2. $\mathrm{r}(M_i) \neq \mathrm{r}(M_1) \Rightarrow n_i = 0$,
3. $n_1 \neq 0$.

We prove ($*$) by induction on $\sum_{i=1}^{m} \mathrm{r}(M_i)$. The context $C(\cdot)$ is the composition of the three contexts $D(\cdot)$, $E(\cdot)$, $F(\cdot)$ defined in the equations (5), (6), (7). The contexts $D(\cdot)$ and $E(\cdot)$ are always in $\Lambda^{\|}$, while $F(\cdot)$ is in $\Lambda^{\|}$ if $\mathbb{M}$ is.

The first context reduces every M_i to an applicative form. Let $\ell \geq 0$ be the maximal length of the prefixes of abstractions of the M_i's and x be a fresh variable, the context

$$D(\cdot) := (\cdot)\,\underbrace{[x^!]\dots[x^!]}_{\ell \text{ times}} \tag{5}$$

reduces each M_i to a onf $\hat{M}_i$ of the form $y_i R_{i,1} \dots R_{i,r_i}$ having the same query of resources as M_i, indeed $\mathrm{r}([x^!]) = 0$.

The second context mainly performs three tasks: it closes every bag $R_{i,1}, \dots, R_{i,r_i}$ (namely, it replaces all the free variables in the bags with closed terms), consequently, the head occurrence of y_i will become the only free occurrence of a variable; the last task consists in gathering the contents of every bag $R_{i,1}, \dots, R_{i,r_i}$ into their product $R_{i,1} \cdot \ \dots\ \cdot R_{i,r_i}$. Let $J := \lambda xz.z[x^!]$ and notice that $JR[J^!]$ reduces to JR for any bag R. Let $\{z_1, \dots, z_h\}$ be the free variables in $\sum_i \hat{M}_i$ (in particular every y_i is in $\{z_1, \dots, z_h\}$), let y be a fresh variable and ℓ be the size of $\sum_i \hat{M}_i$. Define the context

$$E(\cdot) := (\lambda z_1 \dots z_h.(\cdot))\,\underbrace{[\mathbf{X}_\ell{}^!]\dots[\mathbf{X}_\ell{}^!]}_{h \text{ times}}\,\underbrace{[J^!]\dots[J^!]}_{\ell+1 \text{ times}}[y^!]. \tag{6}$$

By Lemma 11 each $R_{i,j}\,\{\mathbf{X}_\ell/z_1\}\dots\{\mathbf{X}_\ell/z_h\}$ reduces to a closed onf $\hat{R}_{i,j}$ such that $\mathrm{r}(\hat{R}_{i,j}) = \mathrm{r}(R_{i,j})$. So we have, denoting the sequence $\{\mathbf{X}_\ell/y_1\}\dots\{\mathbf{X}_\ell/y_m\}$ simply as $\{\mathbf{X}_\ell/\vec{y}\}$:

$$\begin{aligned}
E(y_i R_{i,1} \dots R_{i,r_i}) &\xrightarrow{\mathsf{g}*} \mathbf{X}_\ell(R_{i,1}\,\{\mathbf{X}_\ell/\vec{y}\})\dots(R_{i,r_i}\,\{\mathbf{X}_\ell/\vec{y}\})[J^!]\dots[J^!][y^!]\\
&\xrightarrow{\mathsf{g}*} \mathbf{X}_\ell \hat{R}_{i,1} \dots \hat{R}_{i,r_i}[J^!]\dots[J^!][y^!]\\
&\xrightarrow{\mathsf{g}*} \Big(\lambda x_{r_i+1} \dots x_{\ell+1}.x_{\ell+1} \hat{R}_{i,1} \cdot \ \dots\ \cdot \hat{R}_{i,r_i} \cdot [x^!_{r_i+1}, \dots, x^!_\ell]\Big)[J^!]\dots[J^!][y^!]\\
&\xrightarrow{\mathsf{g}*} J\hat{R}_{i,1} \cdot \ \dots\ \cdot \hat{R}_{i,r_i} \cdot \underbrace{[J^!, \dots, J^!]}_{\ell - r_i}\,\underbrace{[J^!]\dots[J^!]}_{r_i}[y^!]\\
&\xrightarrow{\mathsf{g}*} y\hat{R}_{i,1} \cdot \ \dots\ \cdot \hat{R}_{i,r_i} \cdot [J^!, \dots, J^!].
\end{aligned}$$

To sum up, the composition of $D(\cdot)$ and $E(\cdot)$ gives $E(D(\sum_{i=1}^m M_i)) \xrightarrow{\mathsf{g}*} \sum_{i=1}^m yQ_i$, where for every $i \leq m$, Q_i is a bag of closed terms and $\mathrm{r}(yQ_i) = \mathrm{r}(M_i)$. Notice also that both contexts $D(\cdot)$ and $E(\cdot)$ are in $\Lambda^{\|}$.

Every bag Q_i can be decomposed into a bag $\hat{Q}_i = [L_{i,1}, \dots, L_{i,k_i}]$ of closed linear onf and a bag $\hat{\hat{Q}}_i$ of closed exponentiated simple terms. Of course one or both between $\hat{Q}_i$ and $\hat{\hat{Q}}_i$ can be equal to 1. For every $j \leq k_1$ consider the following family $\mathcal{F}_j$, possibly empty, of pair of indexes:

$$\mathcal{F}_j := \{(i,h) \text{ s.t. } i \leq m,\ h \leq k_i \text{ and } \mathrm{r}(L_{1,j}) \leq \mathrm{r}(L_{i,h})\}$$

Remark that $\sum_{(i,h)\in\mathcal{F}_j} \mathrm{r}(L_{i,h}) < \sum_i \mathrm{r}(M_i)$, hence in case $\mathcal{F}_j$ is non-empty we can apply the induction hypothesis on every $\mathbb{F}_j := \sum_{(i,h)\in\mathcal{F}_j} L_{i,h}$, choosing $L_{1,j}$ as the onf of minimal query of resources. This yields a family of simple outer contexts $\{C_j(\cdot)\}_{j\leq k_1}$ such that for every $(i,h) \in \mathcal{F}_j$,

1. $C_j(L_{i,h}) \xrightarrow{\mathsf{g}*} n_{j,i,h}\mathbf{I}$,
2. $\mathrm{r}(L_{i,h}) \neq \mathrm{r}(L_{1,j}) \Rightarrow n_{j,i,h} = 0$,
3. $n_{j,1,j} \neq 0$.

Actually, since every $L_{i,h}$ is closed one can suppose each $C_j(\cdot)$ to be just applicative, i.e. of the form $(\cdot)P_{j,1} \dots P_{j,p_j}$. In the following, we will denote simply by $\vec{P}_j$ the sequence of bags $P_{j,1} \dots P_{j,p_j}$. Let us define, with z a fresh variable:

$$H := \lambda z.\mathbf{I}[z\vec{P}_1] \dots [z\vec{P}_{k_1}].$$

Notice that whenever $k_1 = 0$ (i.e. Q_1 contains no linear simple term), we do not apply the induction hypothesis and we simply define $H := \lambda z.\mathbf{I}$, which is in $\Lambda^{\|}$. Notice that k_1 is always equal to 0 in the case we started with a $\mathbb{M}$ in $\Lambda^{\|}$.

Let us consider HQ_i for every $i \leq m$. We claim that $HQ_i \xrightarrow{\mathsf{g}*} n_i\mathbf{I}$, with n_1 nonzero and whenever $\mathrm{r}(Q_i) > \mathrm{r}(Q_1)$, $n_i = 0$. Notice that z occurs in H linearly k_1 times and never exponentially. This means that whenever $k_i > k_1$ we have $HQ_i \xrightarrow{\mathsf{g}*} 0$, since H does not succeed in using every linear term in Q_i.

Also in case $k_i < k_1$, we have $HQ_i \xrightarrow{\mathsf{g}*} 0$. Indeed, since $\mathrm{r}(Q_i) \geq \mathrm{r}(Q_1)$, if $k_i < k_1$ there should be a $L_{i,h} \in Q_i$ such that for every $L_{1,j} \in Q_1$, $\mathrm{r}(L_{i,h}) > \mathrm{r}(L_{1,j})$. This means for every $j \leq k_1$, $(i,h) \in \mathcal{F}_j$ and hence $L_{i,h}\vec{P}_j \xrightarrow{\mathsf{g}*} 0$. Thus, denoting by $Q_i/L_{i,h}$ the bag obtained erasing $L_{i,h}$ in Q_i,

$$\begin{aligned} HQ_i &\xrightarrow{\mathsf{b}} \sum_{j=1}^{k_1} \big(\lambda z.\mathbf{I}[z\vec{P}_1] \dots [L_{i,h}\vec{P}_j] \dots [z\vec{P}_{k_1}]\big)Q_i/L_{i,h} \\ &\xrightarrow{\mathsf{g}*} \sum_{j=1}^{k_1} \big(\lambda z.\mathbf{I}[z\vec{P}_1] \dots 0 \dots [z\vec{P}_{k_1}]\big)Q_i/L_{i,h} = 0. \end{aligned}$$

Finally, if $k_i = k_1$, we have $HQ_i \xrightarrow{\mathsf{g}*} \sum_{\sigma \in \mathrm{S}_{k_1}} \mathbf{I}[L_{i,\sigma(1)}\vec{P}_1] \dots [L_{i,\sigma(k_1)}\vec{P}_{k_1}]$. For every permutation $\sigma \in \mathrm{S}_{k_1}$, if for a $j \leq k_1$ we have $\mathrm{r}(L_{i,\sigma(j)}) \neq \mathrm{r}(L_{1,j})$, then there should be $j' \leq k_1$ (possibly $j' = j$) s.t. $\mathrm{r}(L_{i,\sigma(j')}) > \mathrm{r}(L_{1,j'})$, since by hypothesis $\mathrm{r}(M_i) = \sum_{h=1}^{k_i} \mathrm{r}(L_{i,h}) \geq \sum_{h=1}^{k_1} \mathrm{r}(L_{1,h})$ and $k_i = k_1$. We deduce that $L_{i,\sigma(j')}\vec{P}_{j'}$ reduces to 0 and so $\mathbf{I}[L_{i,\sigma(1)}\vec{P}_1] \dots [L_{i,\sigma(k_1)}\vec{P}_{k_1}]$ does. Otherwise, for every $j \leq k_1$, $\mathrm{r}(L_{i,\sigma(j)}) = \mathrm{r}(L_{1,j})$, and so $L_{i,\sigma(j)}\vec{P}_j \xrightarrow{\mathsf{g}*} n_{j,i,\sigma(j)}\mathbf{I}$, for a number $n_{j,i,\sigma(j)}$. We deduce $\mathbf{I}[L_{i,\sigma(1)}\vec{P}_1] \dots [L_{i,\sigma(k_1)}\vec{P}_{k_1}] \xrightarrow{\mathsf{g}*} \big(\prod_{j=1}^{k_1} n_{j,i,\sigma(j)}\big)\mathbf{I}$. In particular, if σ is the identity and $i = 1$, then every $n_{j,i,\sigma(j)}$ is nonzero, and so it is $\prod_{j=1}^{k_1} n_{j,i,\sigma(j)}$. We conclude for an integer n_i:

$$\sum_{\sigma \in \mathrm{S}_{k_1}} \mathbf{I}[L_{i,\sigma(1)}\vec{P}_1] \dots [L_{i,\sigma(k_1)}\vec{P}_{k_1}] \xrightarrow{\mathsf{g}*} n_i\mathbf{I}.$$

In particular, for $i = 1$, n_i is not zero. Moreover, if n_i is nonzero, then there is a permutation $\sigma \in \mathrm{S}_{k_1}$ such that $\prod_{j=1}^{k_1} n_{j,i,\sigma(j)}$ is nonzero. This means that for every $j \leq k_1$ $\mathrm{r}(L_{i,\sigma(j)}) = \mathrm{r}(L_{1,j})$, and so $\mathrm{r}(Q_i) = \mathrm{r}(Q_1)$.

Finally, we define the third context, with x a fresh variable:

$$F(\cdot) := \big(\lambda y.(\cdot)\big)[\lambda x.H[x^!]], \tag{7}$$

$$\frac{}{x:\sigma \vdash_{\mathtt{m}} x:\sigma}\ \mathtt{var} \qquad \frac{}{\vdash_{\mathtt{m}} 1:\omega}\ \mathtt{1} \qquad \frac{\Gamma \vdash_{\mathtt{m}} A_i : \pi}{\Gamma \vdash_{\mathtt{m}} \sum_i A_i : \pi}\ \oplus$$

$$\frac{\Gamma, x:\sigma_1,\ldots,x:\sigma_n \vdash_{\mathtt{m}} M:\tau,\ x \notin d(\Gamma)}{\Gamma \vdash_{\mathtt{m}} \lambda x.M : \sigma_1 \wedge \ldots \wedge \sigma_n \to \tau}\ {\to}\mathtt{I}_n \qquad \frac{\Gamma \vdash_{\mathtt{m}} M : \pi \to \tau \quad \Delta \vdash_{\mathtt{m}} P : \pi}{\Gamma, \Delta \vdash_{\mathtt{m}} MP : \tau}\ {\to}\mathtt{E}$$

$$\frac{\Gamma \vdash_{\mathtt{m}} M : \sigma \quad \Delta \vdash_{\mathtt{m}} P : \pi}{\Gamma, \Delta \vdash_{\mathtt{m}} [M]{\cdot}P : \sigma \wedge \pi}\ \ell \qquad \frac{\Gamma_i \vdash_{\mathtt{m}} M : \sigma_i, \text{ for } 1 \leq i \leq n \quad \Delta \vdash_{\mathtt{m}} P : \pi}{\Gamma_1,\ldots,\Gamma_n,\Delta \vdash_{\mathtt{m}} [M^!]{\cdot}P : \sigma_1 \wedge \ldots \wedge \sigma_n \wedge \pi}\ !_n$$

Figure 4: The type assignment system $\vdash_{\mathtt{m}}$. The rules ${\to}\mathtt{I}_n$ and $!_n$ are parametrized by a natural number n, their 0-ary versions ${\to}\mathtt{I}_0$ and $!_0$ yield $\omega \to \tau$ and π respectively.

and we conclude: $F(yQ_i) \xrightarrow{\mathtt{g}*} HQ_i \xrightarrow{\mathtt{g}*} n_i\mathbf{I}$, where n_i is nonzero in case $i = 1$, and it is zero in case $\mathrm{r}(Q_i) > \mathrm{r}(Q_1)$. □

Notice that linear resources play a crucial role in the above proof. For an example, consider the must-outer normal form $y[x]+y[x,\Omega^!]$. The resource context satisfying Lemma 12 is $C(\cdot) := \big(\lambda yx.(\cdot)\big)[\lambda k.\mathbf{I}[k]]$, indeed $C(y[x]) + C(y[x,\Omega^!])$ reduces to $2\mathbf{I}$. If you consider the λ-context $C^\lambda(\cdot) := \big(\lambda yx.(\cdot)\big)[(\lambda k.\mathbf{I}[k^!])^!]$, we have $C^\lambda(y[x]) + C^\lambda(y[x,\Omega^!]) \xrightarrow{\mathtt{g}*} 2\mathbf{I} + \lambda x.\Omega$.

Theorem 13. *Let M be a term with resources. If M has a may-outer-normal form (resp. must-outer-normal form) then M is may-solvable(resp. must-solvable); moreover, if $M \in \Lambda^{\|}$ then M is may-solvable (resp. must-solvable) in $\Lambda^{\|}$.*

Proof. By applying Lemma 12 to the outer-normal forms in a may/must-outer-normal form of M. □

4 Characterization of May Solvability

In this section we study the converse of Theorem 13 for the "may" semantics: the properties of may-outer normalizability and may-solvability are equivalent in Λ^r and in $\Lambda^{\|}$. The key ingredient we use is an intersection type system $\vdash_{\mathtt{m}}$ assigning types to all and only the expressions having may-outer-normal form, so giving a complete characterization of may-solvability, from the syntactical, operational and logical point of view (Theorem 25).

The system $\vdash_{\mathtt{m}}$ lacks idempotency ($\sigma \wedge \sigma \neq \sigma$): in fact, we use the intersection as logical counterpart of the multiset union. The system has some similarities with that one in [BCL99], which supplies a logical semantics for the language in [Bou93]. The main logical difference between the two systems is that the one in [BCL99] is affine and describes a lazy operational semantics. In the restricted setting of λ-calculus similar non-idempotent systems have been considered starting from [CDCV80], e.g. [Kfo00, WDMT02, NM04, dC09].

Definition 14. *The set of types is the union of the set of* linear types *and that of* intersection types*, given by the following grammars*

$$\begin{aligned} \sigma, \tau &::= a \mid \pi \to \sigma && \textit{linear types} \\ \pi, \zeta &::= \sigma \mid \omega \mid \pi \wedge \zeta && \textit{intersection types} \end{aligned}$$

where a ranges over a countable set of atoms and ω is a constant. We consider types modulo the equivalence $\sim$ generated by the following rules:

$$\pi \wedge \zeta \sim \zeta \wedge \pi, \quad \pi \wedge \omega \sim \pi, \quad \pi_1 \wedge (\pi_2 \wedge \pi_3) \sim (\pi_1 \wedge \pi_2) \wedge \pi_3$$

i.e., $\sim$ states that $\wedge$ defines a commutative monoid with ω as the neutral element. The last two rules allow us to consider n-ary intersections $\sigma_1 \wedge \ldots \wedge \sigma_n$, for any $n \in \mathtt{Nat}$, ω being the 0-ary intersection.

A basis *is a finite multiset of assignments of the shape $x : \sigma$, where x is a variable and σ is a linear type. Capital Greek letters Γ, Δ range over bases. We denote by* $\mathrm{d}(\Gamma)$ *the set of variables occurring in Γ and by Γ, Δ the multiset union between the bases Γ and Δ. A* typing judgement *is a sequent $\Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi$.*

The $\vdash_{\mathtt{m}}$ type assignment system derivates typing judgements for $\mathtt{Nat}\langle\Lambda^{(b)}\rangle$. Its rules are defined in Figure 4. Capital Greek letters Φ, Ψ range over derivations, $\Phi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi$ denoting a derivation Φ with conclusion $\Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi$.

Some comments are in order. The bases have a multiplicative behaviour, and there is no weakening nor contraction, in neither explicit nor implicit form. In the rule $!_n$ the parameter n takes into account the number of times a reusable resource will be called, whereas the rule ℓ assigns just one type to the linear resource M. Duplication and erasure is handled at the level of types by the intersection. For example, in $(\lambda x.y)[M^!]$, $\lambda x.y$ is typed by $\omega \to \sigma$ (by the rules $\mathtt{var}$ and $\to\mathtt{I}_0$), for some σ and $[M^!]$ is typed by ω (by the rules $\mathtt{1}$ and $!_0$), and in $(\lambda x.y[x][x])[y^!]$, $\lambda x.y[x][x]$ is typed by $\sigma \wedge \sigma \to \tau$, using the rule $\to\mathtt{I}_2$, and $[y^!]$ is typed by $\sigma \wedge \sigma$, using rule $!_2$, for any σ and τ.

All other rules are almost standard.

Definition 15. *The **measure** of a derivation Φ is the number* $\mathrm{m}(\Phi)$ *of axioms (i.e.* $\mathtt{var}$ *and* $\mathtt{1}$ *rules) in Φ. The measure* $\mathrm{m}(\mathbb{A})$ *of a sum of expressions $\mathbb{A}$ is*

$$\mathrm{m}(\mathbb{A}) := \inf\{\mathrm{m}(\Phi) \ ; \ \Phi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi, \textit{ for } \Gamma \textit{ a basis and } \pi \textit{ a type}\}.$$

An easy inspection of the rules in Figure 4 will convince the reader that the shape of a type derivation is strictly related with that of the expression it types, as formally stated by the following lemma.

Lemma 16 (Generation)**.**

1. *$\Pi :: \Gamma \vdash_{\mathtt{m}} x : \pi$ implies $\pi = \sigma$ and $\Gamma = x : \sigma$, and Π is an instance of the axiom* $\mathtt{var}$*.*
2. *$\Pi :: \Gamma \vdash_{\mathtt{m}} 1 : \pi$ implies Π is an instance of the axiom* $\mathtt{1}$*, hence $\Gamma = \emptyset$ and $\pi = \omega$.*
3. *$\Pi :: \Gamma \vdash_{\mathtt{m}} \lambda x.M : \pi$ implies $\pi = (\sigma_1 \wedge \ldots \wedge \sigma_n) \to \tau$ and Π is an application of the rule $\to\mathtt{I}_n$ with premise $\Pi' :: \Gamma, x : \sigma_1, \ldots, x : \sigma_n \vdash_{\mathtt{m}} M : \tau$; clearly,* $\mathrm{m}(\Pi) = \mathrm{m}(\Pi')$*.*

4. $\Pi :: \Gamma \vdash_{\mathtt{m}} MP : \pi$ *implies* $\pi = \sigma$, $\Gamma = \Gamma_1, \Gamma_2$ *and* Π *is an application of the rule* $\to$E *with premises* $\Pi_1 :: \Gamma_1 \vdash_{\mathtt{m}} M : \pi' \to \sigma$ *and* $\Pi_2 :: \Gamma_2 \vdash_{\mathtt{m}} P : \pi'$ *for an intersection type* π'*; also,* $\mathrm{m}(\Pi) = \mathrm{m}(\Pi_1) + \mathrm{m}(\Pi_2)$.

5. $\Pi :: \Gamma \vdash_{\mathtt{m}} [M]\cdot P : \pi$ *implies there are two derivations* $\Pi_1 :: \Gamma_1 \vdash_{\mathtt{m}} M : \sigma$ *and* $\Pi_2 :: \Gamma_2 \vdash_{\mathtt{m}} P : \pi'$ *such that* $\pi = \sigma \wedge \pi'$, $\Gamma = \Gamma_1, \Gamma_2$ *and* $\mathrm{m}(\Pi) = \mathrm{m}(\Pi_1) + \mathrm{m}(\Pi_2)$.

6. $\Pi :: \Gamma \vdash_{\mathtt{m}} [M^!]\cdot P : \pi$ *implies there are a number* $n \geq 0$ *of derivations* $\Pi_i :: \Gamma_i \vdash_{\mathtt{m}} M : \sigma_i$*, and a derivation* $\Pi_{n+1} :: \Gamma_{n+1} \vdash_{\mathtt{m}} P : \pi'$ *such that* $\pi = \sigma_1 \wedge \ldots \wedge \sigma_n \wedge \pi'$, $\Gamma = \Gamma_1, \ldots, \Gamma_{n+1}$ *and* $\mathrm{m}(\Pi) = \sum_{i=1}^{n+1} \mathrm{m}(\Pi_i)$.

7. $\Pi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi$ *implies there is an expression* A *in the sum* $\mathbb{A}$ *and a derivation* $\Pi' :: \Gamma \vdash_{\mathtt{m}} A : \pi$ *such that* $\mathrm{m}(\Pi) = \mathrm{m}(\Pi')$.

Notice in the above lemma that the cases of the bag decomposition (items 5, 6) refer to proofs that can be different from the subproofs of Π. This is due to the fact that bags can be broken up in various ways: different decompositions yield typing derivations having some swapped rules ℓ and $!_n$. Also, notice that terms and sums of terms are typed by linear types, bags and sums of bags by intersection types.

The following lemmata (Lemma 17-21) state basically that the typing system behaves well with respect to the substitutions of the resource calculus. They are needed to prove that typing judgements are invariant under baby, middle and giant-step reductions (Proposition 22).

Lemma 17 (Linear Substitution)**.** *Let* $\Phi :: \Gamma, x : \tau \vdash_{\mathtt{m}} \mathbb{A} : \pi$ *and* $\Psi :: \Delta \vdash_{\mathtt{m}} N : \tau$*. There is a derivation* $\mathcal{L}(\Phi, \Psi) :: \Gamma, \Delta \vdash_{\mathtt{m}} \mathbb{A}\langle N/x\rangle : \pi$ *with* $\mathrm{m}(\mathcal{L}(\Phi, \Psi)) = \mathrm{m}(\Phi) + \mathrm{m}(\Psi) - 1$.

Proof. By induction on the structure of the derivation Φ. We treat in detail only the case of a terminal $!_n$ rule. The base of induction is trivial: `var` is immediate, while `1` does not meet the condition of having $x : \tau$ in the basis. The cases $\to\mathtt{I}_n$, $\oplus$ are immediate consequences of the induction hypothesis, the cases $\to$E, ℓ are easier variant of the $!_n$ case. So let us assume

$$\Phi := \dfrac{\begin{matrix}\vdots\, \Phi_i \\ \Gamma_i \vdash_{\mathtt{m}} M : \sigma_i, \quad \text{for } 1 \leq i \leq n\end{matrix} \qquad \begin{matrix}\vdots\, \Phi_P \\ \Gamma_P \vdash_{\mathtt{m}} P : \zeta\end{matrix}}{\Gamma_1, \ldots, \Gamma_n, \Gamma_P \vdash_{\mathtt{m}} [M^!]\cdot P : \sigma_1 \wedge \ldots \wedge \sigma_n \wedge \zeta}\; !_n$$

We suppose the underlined hypothesis $x : \tau$ is in Γ_1, i.e. $\Gamma_1 = \Gamma_1', x : \tau$ (the case $x : \tau$ is in another Γ_i or in Γ_P being similar). Notice that supposing $x : \tau$ in Γ_1 entails $n \geq 1$. By induction there is $\mathcal{L}(\Phi_1, \Psi) :: \Gamma_1', \Delta \vdash_{\mathtt{m}} M\langle N/x\rangle : \sigma_1$ s.t. $\mathrm{m}(\mathcal{L}(\Phi_1, \Psi)) = \mathrm{m}(\Phi_1) + \mathrm{m}(\Psi) - 1$. Let $M\langle N/x\rangle = \sum_{j=1}^{k} L_j$. By Generation (Lemma 16, item 7) $M\langle N/x\rangle$ must have a term (i.e. $k > 0$), say L_1, and a proof $\Phi_1' :: \Gamma_1', \Delta \vdash_{\mathtt{m}} L_1 : \sigma_1$ s.t. $\mathrm{m}(\Phi_1') = \mathrm{m}(\mathcal{L}(\Phi_1, \Psi))$. We define

$$\mathcal{L}(\Phi, \Psi) := \dfrac{\dfrac{\begin{matrix}\vdots\, \Phi_1' \\ \Gamma_1', \Delta \vdash_{\mathtt{m}} L_1 : \sigma_1\end{matrix} \qquad \dfrac{\begin{matrix}\vdots\, \Phi_i \\ \Gamma_i \vdash_{\mathtt{m}} M : \sigma_i, \quad \text{for } 2 \leq i \leq n\end{matrix} \qquad \begin{matrix}\vdots\, \Phi_P \\ \Gamma_P \vdash_{\mathtt{m}} P : \zeta\end{matrix}}{\Gamma_2, \ldots, \Gamma_n, \Gamma_P \vdash_{\mathtt{m}} [M^!]\cdot P : \sigma_2 \wedge \ldots \wedge \sigma_n \wedge \zeta}\; !_{n-1}}{\Gamma_1', \Delta, \Gamma_2, ..., \Gamma_n, \Gamma_P \vdash_{\mathtt{m}} [L_1, M^!]\cdot P : \sigma_1 \wedge ... \wedge \sigma_n \wedge \zeta}\; \ell}{\Gamma_1', \Delta, \Gamma_2, \ldots, \Gamma_n, \Gamma_P \vdash_{\mathtt{m}} \sum_{j=1}^{k} [L_j, M^!]\cdot P : \sigma_1 \wedge ... \wedge \sigma_n \wedge \zeta}\; \oplus$$

where by definition $[M\langle N/x\rangle, M^!]\cdot P = \sum_{j=1}^{k}[L_j, M^!]\cdot P$, and if $k = 1$ the last $\oplus$ rule is omitted. Moreover, $\mathrm{m}(\mathcal{L}(\Phi, \Psi)) = \mathrm{m}(\Phi'_1) + \mathrm{m}(\Phi_P) + \sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\mathcal{L}(\Phi_1, \Psi)) + \mathrm{m}(\Phi_P) + \sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\Phi_1) + \mathrm{m}(\Psi) - 1 + \mathrm{m}(\Phi_P) + \sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\Phi) + \mathrm{m}(\Psi) - 1$. $\square$

Lemma 18 (Linear Expansion)**.** *Let $\Phi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A}\langle N/x\rangle : \pi$. There are a linear type τ and derivations $\Phi_1 :: \Gamma_1, x : \tau \vdash_{\mathtt{m}} \mathbb{A} : \pi$ and $\Phi_2 :: \Gamma_2 \vdash_{\mathtt{m}} N : \tau$ with $\Gamma = \Gamma_1, \Gamma_2$.*

Proof. By structural induction on the sum $\mathbb{A}$. We detail the case $\mathbb{A} = [M^!]\cdot P$, the other cases being easy variants. In this case, $\mathbb{A}\langle N/x\rangle = [M\langle N/x\rangle, M^!]\cdot P + [M^!]\cdot P\langle N/x\rangle$, so Φ types only one term of the sum $\mathbb{A}\langle N/x\rangle$ through a $\oplus$ rule. Let us suppose this term is in $[M\langle N/x\rangle, M^!]\cdot P$ (the case it is in $[M^!]\cdot P\langle N/x\rangle$ being easier), so being of the form $[M', M^!]\cdot P$, with $M\langle N/x\rangle = M' + \mathbb{M}$. By Generation (Lemma 16, items 5 and 6) there is a derivation $\overline{\Phi}^1 :: \Gamma^1 \vdash_{\mathtt{m}} M' : \sigma$ and a derivation $\overline{\Phi}^2 :: \Gamma^2 \vdash_{\mathtt{m}} [M^!]\cdot P : \overline{\pi}$ s.t. $\Gamma = \Gamma^1, \Gamma^2$, $\pi = \sigma \wedge \overline{\pi}$ and $\overline{\Phi}^2$ ends in a $!_n$ rule with n premises typing M and one premise typing P. Possibly applying one $\oplus$ rule to $\overline{\Phi}^1$ we get a derivation of $\Gamma^1 \vdash_{\mathtt{m}} M\langle N/x\rangle : \sigma$, hence by induction hypothesis we have $\overline{\Phi}^1_1 :: \Gamma^1_1, x : \tau \vdash_{\mathtt{m}} M : \sigma$ and $\overline{\Phi}^1_2 :: \Gamma^1_2 \vdash_{\mathtt{m}} N : \tau$. Then we define $\Phi_1 :: \Gamma_1, x : \tau \vdash_{\mathtt{m}} [M^!]\cdot P : \pi$ as a $!_{n+1}$ rule with premise $\overline{\Phi}^1_1$ plus the premises of the $!_n$ rule ending $\overline{\Phi}^2$, and Φ_2 as $\overline{\Phi}^1_2$. $\square$

Lemma 19 (Partial Substitution)**.** *Let $m \geq 0$, $\Phi :: \Gamma, x : \sigma_1, \ldots, x : \sigma_m \vdash_{\mathtt{m}} \mathbb{A} : \pi$ and $\forall i \leq m$, $\Psi_i :: \Delta_i \vdash_{\mathtt{m}} N : \sigma_i$ with $\Delta = \Delta_1, \ldots, \Delta_m$. There is $\mathcal{P}(\Phi, \Psi_{i\leq m}) :: \Gamma, \Delta \vdash_{\mathtt{m}} \mathbb{A}\{(N + x)/x\} : \pi$ with $\mathrm{m}(\mathcal{P}(\Phi, \Psi_{i\leq m})) = \mathrm{m}(\Phi) - m + \sum_{i=1}^{m}\mathrm{m}(\Psi_i)$.*

Proof. Like in the proof of Linear Substitution (Lemma 17), we do induction on the structure of Φ. In the `var` case, we have $\mathbb{A} = y$ and $m = 0$ or 1. If $y \neq x$, then $m = 0$ and we set $\mathcal{P}(\Phi, \Psi_{i\leq m}) = \Phi$. Otherwise, $y = x$, so $\mathbb{A}\{N + x/x\} = N + x$. Hence we define $\mathcal{P}(\Phi, \Psi_{i\leq m})$ by adding one $\oplus$ rule to Φ or to Ψ_1, depending whether $m = 0$ or $m = 1$. In both cases, we easily deduce $\mathrm{m}(\mathcal{P}(\Phi, \Psi_{i\leq m})) = \mathrm{m}(\Phi) - m + \sum_{i=1}^{m}\mathrm{m}(\Psi_i)$. The case `1` is the last rule is immediate.

As for the induction step we detail only the case of a terminal $!_n$ rule, the other cases being immediate or easier variants. So let

$$\Phi := \dfrac{\begin{array}{cc} \vdots\ \Phi_j & \vdots\ \Phi_{n+1} \\ \Gamma_j, \Gamma^x_j \vdash_{\mathtt{m}} M : \tau_j,\ \text{for } 1 \leq j \leq n & \Gamma_{n+1}, \Gamma^x_{n+1} \vdash_{\mathtt{m}} P : \zeta \end{array}}{\Gamma, x : \sigma_1, \ldots, x : \sigma_m \vdash_{\mathtt{m}} [M^!]\cdot P : \tau_1 \wedge \ldots \wedge \tau_n \wedge \zeta}\ !_n$$

where $\Gamma = \Gamma_1, \ldots, \Gamma_{n+1}$ and $x : \sigma_1, \ldots, x : \sigma_m = \Gamma^x_1, \ldots, \Gamma^x_{n+1}$. [4] Notice $\mathrm{m}(\Phi) = \sum_{j=1}^{n+1}\mathrm{m}(\Phi_j)$. For every $j \leq n+1$, let I^j be the set of $i \leq m$ s.t. $x : \sigma_i$ is in Γ^x_j, m^j being the cardinality of I^j, possibly 0. Notice $m = \sum_{j=1}^{n+1} m^j$. Let Δ_{I^j} be the multiset union of the Δ_i bases with $i \in I^j$.We apply the induction hypothesis to each pair Φ_j and $\Psi_{i\in I^j}$, getting a derivation $\mathcal{P}(\Phi_j, \Psi_{i\in I^j}) :: \Gamma_j, \Delta_{I^j} \vdash_{\mathtt{m}} M\{N + x/x\} : \tau_j$ for every $j \leq n$, and $\mathcal{P}(\Phi_{n+1}, \Psi_{I^{n+1}}) :: \Gamma_{n+1}, \Delta_{I^{n+1}} \vdash_{\mathtt{m}} P\{N + x/x\} : \zeta$, such that $\mathrm{m}(\mathcal{P}(\Phi_j, \Psi_{i\in I^j})) = \mathrm{m}(\Phi_j) - m^j + \sum_{i\in I^j}\mathrm{m}(\Psi_i)$ for

[4] Be careful not to confuse m (the number of underlined occurrences of x) with n (the number of premises typing M in the $!_n$ rule).

every $j \leq n+1$. Moreover, in case $m^j = 0$ we have $\mathrm{m}(\mathcal{P}(\Phi_j, \Psi_{i \in I^j})) = \mathrm{m}(\Phi_j)$. Let us shorten the notation, setting $\Delta_{I^j} = \biguplus_{i \in \ I^j} \Delta_i$.

As always, $M\{N + x/x\}$ (resp. $P\{N + x/x\}$) is generally a sum $\sum_{h=1}^{k} M_h$ (resp. $\mathbb{P}$). Let us suppose $k \geq 2$, the case $k = 0$ not holding since $M\{N + x/x\}$ is typed and the case $k = 1$ being immediate. Applying Generation (Lemma 16) to each $\mathrm{m}(\mathcal{P}(\Phi_j, \Psi_{i \in I^j}))$ we obtain a function $f : \{0, \ldots, n-1\} \to \{0, \ldots, k-1\}$, an term P' in the sum $\mathbb{P}$, and a family of derivations $\Phi'_j :: \Gamma_j, \Delta_{I^j} \vdash_{\mathtt{m}} M_{f(j)} : \tau_j$ for $j \leq n$, and $\Phi'_{n+1} :: \Gamma_{n+1}, \Delta_{I^{n+1}} \Delta_i \vdash_{\mathtt{m}} P' : \zeta$, s.t. $\mathrm{m}(\mathcal{P}(\Phi_j, \Psi_{i \in I^j})) = \mathrm{m}(\Phi'_j)$ for $j \leq n+1$. For every $h \leq k$, let $J_h = f^{-1}(h)$, and l_h be the cardinality of J_h; for h, $0 \leq h < k$, let $\pi_0 = \zeta$, $\pi_{h+1} = \pi_h \wedge \bigwedge_{j \in f^{-1}(h+1)} \tau_j$, and $\overline{\Gamma}_0 = \Gamma_{n+1}$, $\overline{\Gamma}_{h+1} = \overline{\Gamma}_h, \Gamma_{f^{-1}(h)}$, and $\overline{\Delta}_0 = \Delta_{I^{n+1}}$, $\overline{\Delta}_{h+1} = \overline{\Delta}_h, \Delta_{\substack{j \in f^{-1}(h) \\ i \in I^j}}$, where, consistency as before, $\Gamma_{f^{-1}(h)}$ (resp. $\Delta_{\substack{j \in f^{-1}(h) \\ i \in I^j}}$) denotes the multiset union of the Γ_j (resp. Δ_i) bases with $j \in f^{-1}(h)$ (resp. $i \in \bigcup_{j \in f^{-1}(h)} I^j$). Recalling $\pi_k = \tau_1 \wedge \ldots \wedge \tau_n \wedge \zeta = \pi$, we have $\mathcal{P}(\Phi, \Psi_{i \leq m}) :=$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ \Phi'_j}{\Gamma_j, \Delta_{I^j} \vdash_{\mathtt{m}} M_k : \tau_k, \text{ for } j \in J_k} \qquad \dfrac{\dfrac{\dfrac{\vdots\ \Phi'_j}{\Gamma_j, \Delta_{I^j} \vdash_{\mathtt{m}} M_1 : \tau_j, \text{ for } j \in J_1} \quad \dfrac{\vdots\ \Phi'_{n+1}}{\Gamma_{n+1}, \Delta_{I^{n+1}} \vdash_{\mathtt{m}} P' : \zeta}}{\overline{\Gamma}_1, \overline{\Delta}_1 \vdash_{\mathtt{m}} [M_1^!] \cdot P' : \pi_1}\ !_{l_1}}{\overline{\Gamma}_{k-1}, \overline{\Delta}_{k-1} \vdash_{\mathtt{m}} [M_1^!, \ldots, M_{k-1}^!] \cdot P' : \pi_{k-1}}\ \vdots}{\Gamma, \Delta \vdash_{\mathtt{m}} [(M\{N + x/x\})^!] \cdot P' : \tau_1 \wedge \ldots \wedge \tau_n \wedge \zeta}\ !_{l_k}}{\Gamma, \Delta \vdash_{\mathtt{m}} [(M\{N + x/x\})^!] \cdot \mathbb{P} : \tau_1 \wedge \ldots \wedge \tau_n \wedge \zeta}\ \oplus$$

We have $\mathrm{m}(\mathcal{P}(\Phi, \Psi_{i \leq m})) = \sum_{j=1}^{n+1} \mathrm{m}(\phi'_j) = \sum_{j=1}^{n+1} \mathrm{m}(\mathcal{P}(\Phi_j, \Psi_{i \in I^j})) = \sum_{j=1}^{n+1} \big(\mathrm{m}(\Phi_j) + \sum_{i \in I^j} \mathrm{m}(\Psi_i)\big) - \sum_{j=1}^{n+1} m^j = \mathrm{m}(\Phi) - m + \sum_{i=1}^{m} \mathrm{m}(\Psi_i)$. $\square$

The next lemmata have proofs similar to the previous Lemma 17-19 (by induction on the structure of $\mathbb{A}$ or Φ). We omit their proofs.

Lemma 20 (Partial Expansion)**.** *Let $\Phi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A}\{N + x/x\} : \pi$, then there is a number $m \geq 0$, linear types $\tau_1, \ldots, \tau_m$ and derivations $\Phi_1 :: \Gamma_1, x : \tau_1, \ldots, x : \tau_m \vdash_{\mathtt{m}} \mathbb{A} : \pi$ and $\Psi_i :: \Delta_i \vdash_{\mathtt{m}} N : \tau_i$ for $i \leq m$ and $\Gamma = \Gamma_1, \Delta_1, \ldots, \Delta_m$.*

Lemma 21. *Let $x \notin d(\Gamma)$, then for every $\Phi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A} : \pi$ there is $\Psi :: \Gamma \vdash_{\mathtt{m}} \mathbb{A}\{0/x\} : \pi$ with $\mathrm{m}(\Phi) = \mathrm{m}(\Psi)$, and vice versa.*

The following proposition states that the type system $\vdash_{\mathtt{m}}$ enjoys both subject reduction and subject expansion.

Proposition 22 (Invariance of $\vdash_{\mathtt{m}}$ typings)**.** *Let $\epsilon \in \{\mathtt{b}, \mathtt{m}, \mathtt{g}\}$ and $M \xrightarrow{\epsilon} \mathbb{M}$, then M and $\mathbb{M}$ share the same judgements, i.e. $\Gamma \vdash_{\mathtt{m}} M : \tau$ iff $\Gamma \vdash_{\mathtt{m}} \mathbb{M} : \tau$. Also, $M \xrightarrow{\mathtt{og}} \mathbb{M}$ entails $\mathrm{m}(M) > \mathrm{m}(\mathbb{M})$.*

Proof. The proof is by induction on the context enclosing the redex reduced in $M \xrightarrow{\epsilon} \mathbb{M}$. All induction steps follow by induction, taking into account that, whenever the redex is inside a reusable resource $N^!$ (so the reduction is not outer) the measure m may not decrease since (the bag containing) $N^!$ may be typed by ω.

The base of induction is when M is the redex fired by the reduction $M \xrightarrow{\epsilon} \mathbb{M}$. One can consider only the baby-step cases, the giant and the middle ones will

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\Psi_1\\ \Gamma_1, x:\tau, \vec{x}:\vec{\tau} \vdash_{\mathtt{m}} L:\sigma\end{matrix}}{\Gamma_1 \vdash_{\mathtt{m}} \lambda x.L : \tau\wedge\zeta \to \sigma}\,{\to}\mathtt{I}_n \qquad \dfrac{\begin{matrix}\vdots\,\Psi_2\\ \Gamma_2 \vdash_{\mathtt{m}} N:\tau\end{matrix} \quad \begin{matrix}\vdots\,\Psi_3\\ \Gamma_3 \vdash_{\mathtt{m}} P:\zeta\end{matrix}}{\Gamma_2,\Gamma_3 \vdash_{\mathtt{m}} [N]\cdot P : \tau\wedge\zeta}\,\ell}{\Gamma \vdash_{\mathtt{m}} (\lambda x.L)[N]\cdot P:\sigma}\,{\to}\mathtt{E}$$

(a) definition of Ψ

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\,\Phi_1\\ \Gamma_1,\Gamma_2,\vec{x}:\vec{\tau} \vdash_{\mathtt{m}} \overline{L}:\sigma\end{matrix}}{\Gamma_1,\Gamma_2 \vdash_{\mathtt{m}} \lambda x.\overline{L} : \zeta\to\sigma}\,{\to}\mathtt{I}_n \qquad \begin{matrix}\vdots\,\Phi_3\\ \Gamma_3 \vdash_{\mathtt{m}} P:\zeta\end{matrix}}{\Gamma \vdash_{\mathtt{m}} (\lambda x.\overline{L})P:\sigma}\,{\to}\mathtt{E}}{\Gamma \vdash_{\mathtt{m}} (\lambda x.L\langle N/x\rangle)P:\sigma}\,\oplus$$

(b) definition of Φ

Figure 5: Definition of the derivations Ψ and Φ used in the proof of Proposition 22 (case 2).

follow since they correspond to a sequence of baby-steps. In particular, one proves that the measure m is monotone strictly decreasing on every baby-step but the one choosing a bang element from the bag, in which case m is monotone decreasing (case 3 below). Then m strictly decreases on giant-steps since they correspond to sequences of baby-steps ending always in an empty bag baby-step.

Case 1 (empty bag)**.** Let $M = (\lambda x.L)1 \xrightarrow{\mathtt{b}} L\{0/x\} = \mathbb{M}$, and suppose $\Psi :: \Gamma \vdash_{\mathtt{m}} (\lambda x.L)1 : \sigma$. By Generation (Lemma 16) Ψ ends in a $\to\mathtt{E}$ rule with premises $\Psi' :: \Gamma \vdash_{\mathtt{m}} \lambda x.L : \omega \to \sigma$ and an instance of the $\mathtt{1}$ rule, and $\mathrm{m}(\Psi) = \mathrm{m}(\Psi') + 1$. Again by Generation Ψ' ends in a $\to\mathtt{I}_0$ rule with premise $\Psi'' :: \Gamma \vdash_{\mathtt{m}} L : \sigma$ and $x \notin d(\Gamma)$, $\mathrm{m}(\Psi'') = \mathrm{m}(\Psi')$. By Lemma 21 there is a derivation $\Phi :: \Gamma \vdash_{\mathtt{m}} L\{0/x\} : \sigma$ with $\mathrm{m}(\Phi) = \mathrm{m}(\Psi'') = \mathrm{m}(\Psi) - 1$. We conclude that the judgements of $(\lambda x.L)1$ are also of $L\{0/x\}$ and $\mathrm{m}((\lambda x.L)1) \geq \mathrm{m}(L\{0/x\}) + 1$.

Conversely, suppose $\Phi :: \Gamma \vdash_{\mathtt{m}} L\{0/x\} : \sigma$. By definition of substitution, $L\{0/x\}$ is either 0 or a term with no free occurrence of x. The first hypothesis cannot hold, being $L\{0/x\}$ typable, hence $L\{0/x\}$ is a simple term having no free occurrence of x, so $x \notin d(\Gamma)$. Then we can apply Lemma 21, from which we deduce $\Psi :: \Gamma \vdash_{\mathtt{m}} L : \sigma$. Adding to Ψ one $\to\mathtt{I}_0$ rule, and then composing it with one $\mathtt{1}$ rule through a $\to\mathtt{E}$ yields a derivation $\Psi' :: \Gamma \vdash_{\mathtt{m}} (\lambda x.L)1 : \sigma)$. We thus conclude that $(\lambda x.L)1$ and $L\{0/x\}$ share the same types and $\mathrm{m}((\lambda x.L)1) > \mathrm{m}(L\{0/x\})$.

Case 2 (bag with linear resource)**.** Let $M = (\lambda x.L)[N]\cdot P \xrightarrow{\mathtt{b}} (\lambda x.L\langle N/x\rangle)P = \mathbb{M}$ and suppose $\Psi :: \Gamma \vdash_{\mathtt{m}} (\lambda x.L)[N]\cdot P : \sigma$. By Generation (Lemma 16) we can assume Ψ to be as in Figure 5(a), where by $\vec{x} : \vec{\tau}$ we are meaning $x : \tau_1, \ldots, x : \tau_m$, and ζ is $\tau_1 \wedge \ldots \wedge \tau_m$ (in case $m = 0$, $\zeta = \omega$). By Linear Substitution (Lemma 17) we get $\mathcal{L}(\Psi_1, \Psi_2) :: \Gamma_1, \vec{x} : \vec{\tau}, \Gamma_2 \vdash_{\mathtt{m}} L\langle N/x\rangle : \sigma$, with $\mathrm{m}(\mathcal{L}(\Psi_1, \Psi_2)) = \mathrm{m}(\Psi_1) + \mathrm{m}(\Psi_2) - 1$. As usual, we should notice that $L\langle N/x\rangle : \sigma$ might not be a simple term but a sum: in that case $\mathcal{L}(\Psi_1, \Psi_2)$ ends in a $\oplus$ rule with premise a derivation $\Phi_1 :: \Gamma_1, \vec{x} : \vec{\tau}, \Gamma_2 \vdash_{\mathtt{m}} \overline{L} : \sigma$ with $\overline{L}$ a simple term in the sum $L\langle N/x\rangle$ and $\mathrm{m}(\Phi_1) = \mathrm{m}(\mathcal{L}(\Psi_1, \Psi_2))$. Then we define Φ as in Figure 5(b), with $\Phi_3 = \Psi_3$.

We remark that $\mathrm{m}(\Phi) = \mathrm{m}(\Phi_1) + \mathrm{m}(\Phi_3) = \mathrm{m}(\mathcal{L}(\Psi_1, \Psi_2)) + \mathrm{m}(\Psi_3) = \mathrm{m}(\Psi) - 1$. We conclude that every type of $(\lambda x.L)[N]\cdot P$ is also a type of $(\lambda x.L\langle N/x\rangle)P$ and $\mathrm{m}((\lambda x.L)[N]\cdot P) \geq \mathrm{m}((\lambda x.L\langle N/x\rangle)P) + 1$.

Conversely, assume $\Phi :: \Gamma \vdash_{\mathtt{m}} (\lambda x.L\langle N/x\rangle)P : \sigma$. We can suppose Φ as in Figure 5(b), where as above $\vec{x} : \vec{\tau}$ denotes the basis $x : \tau_1, \ldots, x : \tau_m$ with $\zeta = \tau_1 \wedge \ldots \wedge \tau_m$, and in case $L\langle N/x\rangle$ is a simple term the terminal $\oplus$ rule is omitted. By possibly adding one $\oplus$ rule to Φ_1 one get $\overline{\Phi}_1 :: \Gamma_1, \Gamma_2, \vec{x} : \vec{\tau} \vdash_{\mathtt{m}} L\langle N/x\rangle : \sigma$. Applying Linear Expansion (Lemma 18) we have $\Psi_1 :: \Gamma_1, \vec{x} : \vec{\tau}, x : \tau \vdash_{\mathtt{m}} L : \sigma$

and $\Psi_2 :: \Gamma_2 \vdash_{\mathtt{m}} N : \tau$ (where we recall $x \notin \mathrm{FV}(N)$). Then we set Ψ as in Figure 5(a). This proves that the types of $(\lambda x.L\langle N/x\rangle)P$ are also of $(\lambda x.L)[N]{\cdot}P$.

Case 3 (bag with exponential resource)**.** Suppose $M = (\lambda x.L)[N^!] \cdot P \xrightarrow{\mathtt{b}} (\lambda x.L\,\{N+x/x\})P = \mathbb{M}$ and assume $\Psi :: \Gamma \vdash_{\mathtt{m}} (\lambda x.L)[N^!] \cdot P : \sigma$. As in the previous case, Generation (Lemma 16) allows us to suppose

$$\Psi = \dfrac{\dfrac{\begin{array}{c}\vdots\ \Psi_1\\ \Gamma_1, x:\tau_1, ..., x:\tau_n, \vec{x}:\vec{\sigma} \vdash_{\mathtt{m}} L : \sigma\end{array}}{\Gamma_1 \vdash_{\mathtt{m}} \lambda x.L : (\bigwedge_{i=1}^n \tau_i) \wedge \zeta \to \sigma}{\to}\mathtt{I}_n \quad \dfrac{\begin{array}{c}\vdots\ \Psi^i\\ \Gamma_2^i \vdash_{\mathtt{m}} N : \tau_i \text{ for } i \leq n\end{array} \quad \begin{array}{c}\vdots\ \Psi_3\\ \Gamma_3 \vdash_{\mathtt{m}} P : \zeta\end{array}}{\Gamma_2, \Gamma_3 \vdash_{\mathtt{m}} [N^!]{\cdot}P : (\bigwedge_{i=1}^n \tau_i) \wedge \zeta}\ell}{\Gamma \vdash_{\mathtt{m}} (\lambda x.L)[N^!]{\cdot}P : \sigma}{\to}\mathtt{E}$$

where $\Gamma_2 = \Gamma_2^1, \dots, \Gamma_2^n$, and as usual $\vec{x} : \zeta$ denotes the context $x : \sigma_1, \dots, x : \sigma_m$ for $\zeta = \sigma_1 \wedge \dots \wedge \sigma_m$. By Partial Substitution (Lemma 19) we get $\mathcal{P}(\Psi_1, \Psi^{i \leq n}) :: \Gamma_1, \vec{x} : \vec{\sigma}, \Gamma_2 \vdash_{\mathtt{m}} L\,\{N+x/x\} : \sigma$, with $\mathrm{m}(\mathcal{P}(\Psi_1, \Psi^{i\leq n})) = \mathrm{m}(\Psi_1) - n + \sum_{i=1}^n \mathrm{m}(\Psi^i)$. Let us suppose $L\,\{N+x/x\} : \sigma$ is a sum of terms (the case it is a single term being an easier variant): under this hypothesis Generation (Lemma 16) states that $\mathcal{P}(\Psi_1, \Psi^{i\leq n})$ ends in a $\oplus$ rule with premise a derivation $\overline{\Psi} :: \Gamma_1, \vec{x} : \vec{\sigma}, \Gamma_2 \vdash_{\mathtt{m}} \overline{L} : \sigma$ with $\overline{L}$ a simple term in the sum $L\,\{N+x/x\}$ and $\mathrm{m}(\overline{\Psi}) = \mathrm{m}(\mathcal{P}(\Psi_1, \Psi^{i\leq n}))$. Then we define

$$\Phi := \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ \overline{\Psi}\\ \Gamma_1, \vec{x}:\vec{\sigma}, \Gamma_2 \vdash_{\mathtt{m}} \overline{L} : \sigma\end{array}}{\Gamma_1, \Gamma_2 \vdash_{\mathtt{m}} \lambda x.\overline{L} : \zeta \to \sigma}{\to}\mathtt{I}_n \quad \begin{array}{c}\vdots\ \Psi_3\\ \Gamma_3 \vdash_{\mathtt{m}} P : \zeta\end{array}}{\Gamma \vdash_{\mathtt{m}} (\lambda x.\overline{L})P : \sigma}{\to}\mathtt{E}}{\Gamma \vdash_{\mathtt{m}} (\lambda x.L\,\{N+x/x\})P : \sigma}\oplus$$

We remark that $\mathrm{m}(\Phi) = \mathrm{m}(\overline{\Psi}) + \mathrm{m}(\Psi_3) = \mathrm{m}(\mathcal{P}(\Psi_1, \Psi^{i\leq n})) + \mathrm{m}(\Psi_3) = \mathrm{m}(\Psi) - n$. In particular, notice that whenever $n = 0$ the measure is constant.

For the converse we use a similar variant of the above case 2, using Exponential Expansion (Lemma 20) instead of Linear Expansion. We conclude that $(\lambda x.L)[N]{\cdot}P$ and $(\lambda x.L\langle N/x\rangle)P$ share the same judgements and $\mathrm{m}((\lambda x.L)[N^!]{\cdot}P) \geq \mathrm{m}((\lambda x.L\,\{N+x/x\})P)$.

□

We prove the equivalence among may-solvability, typability in $\vdash_{\mathtt{m}}$ and may-outer normalizability (Theorem 25). As a byproduct we achieve also an operational characterization through the notion of outer reduction (Definition 7). In various λ-calculi the implication *typable* $\Rightarrow$ *head-normalizable* is often proven using suitable notions of computability or saturated sets (e.g. [Kri93], and our proof of the Theorem 42), whereas the implication *solvable* $\Rightarrow$ *head-normalizable* is argued through a standardization theorem (e.g. [Bar84]). Our proof is instead based on a different method, namely both implications are easy consequences of Lemma 23, which is argued by induction on the measure on the type derivations given in Definition 15. In the λ-calculus setting, a somewhat similar approach can be found in [Val01]. More generally, the idea of measuring quantitative properties of terms using non-idempotent intersection types can be found also in [dC09], springing out from the analysis of the β-reduction via the notion of Taylor expansion [ER06, ER08]. As an aside let us point out [dCPTdF08] also, where similar methods have been used to study (outer-)normalizability of linear logic proof-nets.

Lemma 23. *Let M be a resource term and $C(\cdot)$ be a simple outer-context. If $C(M)$ is typable, then M is reducible to a* monf *by outer reduction.*

Proof. By induction on $\mathrm{m}(C(M))$, which is a finite number, $C(M)$ being typable. If M is a monf we are done. Otherwise, it has an outer redex, so let $M \xrightarrow{\mathtt{og}} \mathbb{M}$. Since $C(\cdot)$ is a outer-context, every outer redex of M is outer in $C(M)$, hence we have $C(M) \xrightarrow{\mathtt{og}} C(\mathbb{M})$. By Proposition 22 $\mathrm{m}(C(M)) > \mathrm{m}(C(\mathbb{M}))$. Let $\mathbb{M} = M' + \mathbb{M}''$ be such that $\mathrm{m}(C(M')) = \mathrm{m}(C(\mathbb{M}))$: the fact that M' exists is due to $C(\cdot)$ being simple, as every term in the sum $C(\mathbb{M})$ is obtained by plugging a term of $\mathbb{M}$ in $C(\cdot)$. By induction hypothesis M' is outer reducible to a monf $\mathbb{L}$. We conclude by context closure: $M \xrightarrow{\mathtt{og}} M' + \mathbb{M}'' \xrightarrow{\mathtt{og}*} \mathbb{L} + \mathbb{M}''$. $\square$

Lemma 24. *Every may-outer-normal form is typable in $\vdash_{\mathtt{m}}$.*

Proof. By induction on the structure of a monf $\mathbb{M}$. The only interesting case is when $\mathbb{M}$ is of the form $xP_1 \dots P_p$ with each P_i of the form $[M_{i,1}, \dots, M_{i,m_i}] \cdot [N^!_{i,1}, \dots, N^!_{i,n_i}]$, with $m_i, n_i \geq 0$ and for each $j \leq m_i$, M_{j,m_i} onf. By induction hypothesis we have derivations $\Psi_{i,j} :: \Gamma_{i,j} \vdash_{\mathtt{m}} M_{i,j} : \tau_{i,j}$ for each $i \leq p$, $j \leq m_i$ hence we can construct a derivation $\Phi_i :: \Gamma_{i,1}, \dots, \Gamma_{i,m_i} \vdash_{\mathtt{m}} P_i : \tau_{i,1} \wedge \dots \wedge \tau_{i,m_i}$ by applying a tree of m_i rules ℓ having as premises the $\Psi_{i,1}, \dots, \Psi_{i,m_i}$ respectively and, as the rightmost leaf, a derivation of $\vdash_{\mathtt{m}} [N^!_{i,1}, \dots, N^!_{i,n_i}] : \omega$ made of n_i rules $!_0$ and one rule $\mathtt{1}$. Similarly we get a derivation typing $xP_1 \dots P_p$ by applying a tree of p rules $\to\mathtt{E}$ having as premises the Φ_i's derivations and, as the leftmost leaf, a $\mathtt{var}$ rule typing x with $(\bigwedge_{j \leq m_1} \tau_{1,j}) \wedge \dots \wedge (\bigwedge_{j \leq m_p} \tau_{p,j}) \to \sigma$, for a linear type σ. $\square$

Theorem 25. *Given a resource term $M \in \Lambda^r$, the following are equivalent:*

1. *M is may-outer normalizable,*
2. *M is typable by $\vdash_{\mathtt{m}}$,*
3. *M is reducible to a* monf *by outer reduction,*
4. *M is may-solvable.*

Moreover, if $M \in \Lambda^{\|}$ then 4 can be replaced by

4'. M is may-solvable in $\Lambda^{\|}$.

Proof. $1 \Rightarrow 2$: by Proposition 22 and Lemma 24. $2 \Rightarrow 3$: by Lemma 23, merely taking the hole as the simple head-context. $3 \Rightarrow 4$ (resp. $3 \Rightarrow 4'$): by Theorem 13 and context closure. $4 \Rightarrow 1$: if there is a simple head-context $C(\cdot)$ s.t. $C(M)$ has a monf, by the already proven implication $1 \Rightarrow 2$, $C(M)$ is typable, we conclude by Lemma 23. $\square$

The implication $1 \Rightarrow 3$ can also be argued as a corollary of the standardization proven in [PT09]. However, our proof uses the type assignment system, namely Lemma 23, so it adopts a different approach with respect to the techniques in [PT09].

We can now trust in the examples 1-4 of Section 3. In fact $\mathbf{\Omega}$, having no monf, is may (hence must) unsolvable.

5 The problem of Must Solvability

The characterization of the must solvability in Λ^r is a tough problem. In Λ^r we have the empty sum (i.e. 0) and diverging terms (like $\mathbf{\Omega}$). They are both considered unsolvable since they do not communicate with the environment, but they have a crucial difference: 0 is the neutral element of the sum, hence it disappears when added to a solvable term, $\mathbf{\Omega}$ does not. This difference is harmless when considering may-solvability but becomes relevant for the must-solvability. Consider the terms:

$$M_1 = \lambda x.\mathbf{I}[\mathbf{\Omega}^!, x^!] \xrightarrow{\mathrm{g}} \lambda x.\mathbf{\Omega} + \lambda x.x$$
$$M_2 = \lambda x.\mathbf{I}[(\mathbf{\Omega}[x])^!, x^!] \xrightarrow{\mathrm{g}} \lambda x.\mathbf{\Omega}[x] + \lambda x.x$$

We claim that M_1 is must-solvable whilst M_2 is not. In fact, both $\lambda x.\mathbf{\Omega}$ and $\lambda x.\mathbf{\Omega}[x]$ have no monf, so they are not may solvable (Theorem 25). Hence the only possibility for having a sum of identities from M_1 (resp. M_2) is to find a context reducing $\lambda x.\mathbf{\Omega}$ (resp. $\lambda x.\mathbf{\Omega}[x]$) to 0 and keeping $\lambda x.x$ different from 0. Such a context exists for M_1, e.g. $(\cdot)[\mathbf{I}]$, but not for M_2, both $\lambda x.\mathbf{\Omega}[x]$ and $\lambda x.x$ using x linearly.

This example shows: first, that the implication *must-solvable* $\Rightarrow$ *must-outer normalizable*, i.e. the converse of Theorem 13 for the must case, does not hold in Λ^r, M_1 being a contra-example; second, that in order to characterize whether a sum is must-solvable, one should know if there are contexts reducing the diverging terms of a sum to 0 and keeping at least one among the others different from 0. With respect to this point the situation can be even worse, consider:

$$M_3 = \lambda x.\mathbf{I}[(\mathbf{\Omega}[x])^!, y^!] \xrightarrow{\mathrm{g}} \lambda x.\mathbf{\Omega}[x] + \lambda x.y$$

The term M_3 is must-solvable via the context $\lambda y.(\cdot)1$. This means that the term $\lambda x.\mathbf{\Omega}[x]$, occurring in the sums that are reduct of both M_2 and M_3, has a different behaviour depending on the shape of the other terms of the sum. In particular, the property of being must-solvable does not commute with the sum.

Such kinds of problems are too difficult for us at the present, so we study the must-solvability in the simpler setting of $\Lambda^{\|}$, the purely non-deterministic fragment of Λ^r. In this fragment we have only non-empty bags of reusable resources and non-empty sums, so no term reduces to 0, and the problem of must-solvability becomes smoother than in Λ^r.

5.1 Characterization of Must Solvability in $\Lambda^{\|}$

From now on we consider $\Lambda^{\|}$, the purely non-deterministic fragment of Λ^r that has been defined at the end of Section 2.

A type assignment system characterizing logically the must-solvability must assign types to all the elements of a sum, so to type the sum itself (by definition of must-solvability), and, in order to type an application $(\lambda x.M)P$ the types of all elements of P need to match the types of all the free occurrences of x in M. Namely, the type assignment needs to have a correct behaviour with respect to the λ-calculus, which is a proper fragment of $\Lambda^{\|}$, i.e., a term of Λ needs to be typed in it if and only if it has head-normal form, in the standard sense.

Therefore, we use a well-know λ-model, namely the D_∞ of Scott [Sco76], described in logical form in [PRDR04]. In fact, the underlined structure is a

$$\pi \preceq \omega \qquad \pi \preceq \pi \wedge \pi \qquad \pi_1 \wedge \pi_2 \preceq \pi_i \ (i = 1, 2) \qquad \omega \to \phi \preceq \phi \preceq \omega \to \phi$$

$$\omega \preceq \omega \to \omega \qquad \pi \to \omega \preceq \omega \to \omega \qquad (\pi \to \rho) \wedge (\pi \to \zeta) \preceq \pi \to (\rho \wedge \zeta)$$

$$\frac{\pi \preceq \pi' \quad \rho \preceq \rho'}{\pi \wedge \rho \preceq \pi' \wedge \rho'} \qquad \frac{\pi' \preceq \pi \quad \rho \preceq \rho'}{\pi \to \rho \preceq \pi' \to \rho'}$$

(a) rules defining the preorder $\preceq$ over types

$$\frac{}{\Gamma, x : \pi \vdash_{\mathtt{M}} x : \pi}\,\mathtt{var} \qquad \frac{}{\Gamma \vdash_{\mathtt{M}} M : \omega}\,\omega \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : \pi \quad \Gamma \vdash_{\mathtt{M}} \mathbb{N} : \pi}{\Gamma \vdash_{\mathtt{M}} M + \mathbb{N} : \pi}\,\mathtt{mix+}$$

$$\frac{\Gamma, x : \pi \vdash_{\mathtt{M}} M : \rho}{\Gamma \vdash_{\mathtt{M}} \lambda x.M : \pi \to \rho}\,\to\mathtt{I} \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : \pi \to \tau \quad \Gamma \vdash_{\mathtt{M}} P : \pi}{\Gamma \vdash_{\mathtt{M}} MP : \tau}\,\to\mathtt{E}$$

$$\frac{\Gamma \vdash_{\mathtt{M}} M : \pi}{\Gamma \vdash_{\mathtt{M}} [M^!] : \pi}\,! \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : \pi \quad \Gamma \vdash_{\mathtt{M}} P : \pi}{\Gamma \vdash_{\mathtt{M}} [M^!]{\cdot}P : \pi}\,\mathtt{mix}\cdot$$

$$\frac{\Gamma \vdash_{\mathtt{M}} M : \pi \quad \Gamma \vdash_{\mathtt{M}} M : \rho}{\Gamma \vdash_{\mathtt{M}} M : \pi \wedge \rho}\,\wedge I \qquad \frac{\Gamma \vdash_{\mathtt{M}} M : \rho \quad \rho \preceq \pi}{\Gamma \vdash_{\mathtt{M}} M : \pi}\,\preceq$$

(b) deductive rules of $\vdash_{\mathtt{M}}$

Figure 6: The type assignment system $\vdash_{\mathtt{M}}$.

complete lattice with infinite descending chains, so every pair of compact points has a least upper bound (lub) greater than $\perp$ and a greatest lower bound (glb) less than $\top$. From a logical point of view, since types are names for compact elements, this means that every two typable terms have a non-trivial type in common. Since D_∞ is a sensible model, giving non-trivial interpretation to all and only the solvable terms, this implies that every pair of solvable terms have at least one type in common.

The type assignment system $\vdash_{\mathtt{M}}$ characterizing must-solvability is exactly the logical description of D_∞, plus two new rules for typing bags and sums. Note that, thanks to the previously described property, these rules are very easy, since they ask that all the elements of a bag or a sum have the same type.

Definition 26. *The set of types for $\vdash_{\mathtt{M}}$ is given by the following grammar:*

$$\pi, \rho, \zeta ::= \phi \mid \omega \mid \pi \to \rho \mid \pi \wedge \rho$$

where ϕ and ω are two different constants. We consider the smallest pre-order relation $\preceq$ over types satisfying the rules in Figure 6(a). The relation $\approx$ is the equivalence generated by $\preceq$. The symbol $=$ denote the identity relation on types.

A basis *is a function from variables to types, with finite domain. A* typing judgement *is a sequent $\Gamma \vdash_{\mathtt{M}} \mathbb{A} : \pi$, where Γ is a basis. The rules of the type assignment system $\vdash_{\mathtt{M}}$ are defined in Figure 6(b).*

We extend to $\vdash_{\mathtt{M}}$ the notations used in the previous section for $\vdash_{\mathtt{m}}$.

The system $\vdash_{\mathtt{M}}$ is very different from $\vdash_{\mathtt{m}}$. Essentially, $\vdash_{\mathtt{m}}$ has a quantitative aspect missing in $\vdash_{\mathtt{M}}$. This is reflected in the fact that the intersection is idempotent in the latter and not in the former. Indeed, in $\vdash_{\mathtt{m}}$ the non-idempotency of the intersection is crucial to account for the consumption of linear resources, while in $\vdash_{\mathtt{M}}$ we do not need to count the number of resources in a bag since $\Lambda^{\|}$ allows only reusable resources. Moreover, in order to type a sum, all elements of the sum need to be typed by the same type. The constant type ω represents the empty bag in $\vdash_{\mathtt{m}}$, while in $\vdash_{\mathtt{M}}$ it is a trivial type, which can be assigned to all terms. Indeed, the empty bag does not belong to $\Lambda^{\|}$. Another difference is that in $\vdash_{\mathtt{M}}$ we allow intersection in the right side of an arrow, but this is made for making the type matching exactly the compact elements of D_∞. Since $\pi_1 \wedge \pi_2 \preceq \pi_i$, the rule $\preceq$ entails the $\wedge$-elimination rule, which we have therefore omitted in the definition of $\vdash_{\mathtt{M}}$:

$$\frac{\Gamma \vdash_{\mathtt{M}} M : \pi_1 \wedge \pi_2}{\Gamma \vdash_{\mathtt{M}} M : \pi_i} \wedge E$$

Also the following rules are admissible:

$$\frac{\Gamma \vdash_{\mathtt{M}} M : \rho \quad x \notin d(\Gamma)}{\Gamma, x : \pi \vdash_{\mathtt{M}} M : \rho} \mathtt{weak} \qquad \frac{\Gamma, x : \zeta \vdash_{\mathtt{M}} M : \rho \quad \pi \preceq \zeta}{\Gamma, x : \pi \vdash_{\mathtt{M}} M : \rho} \preceq\mathtt{L}$$

The following lemma states well-known properties of $\approx$ that we will use in the sequel. We refer to [RDRP04] (chapter 11.1.1, p. 132) for its proof.

Lemma 27 (Properties of $\preceq$, [RDRP04])**.** *The constants ω and ϕ belong to the $\approx$-classes that are, respectively, the $\preceq$-maximum and the $\preceq$-minimum. Indeed such classes can be characterized as follows:*

$$\begin{array}{ll} & \phi \not\approx \omega \\ \pi \wedge \rho \approx \omega \ \textit{ iff } \ \pi \approx \omega \textit{ and } \rho \approx \omega, & \pi \wedge \rho \approx \phi \ \textit{ iff } \ \pi \approx \phi \textit{ or } \rho \approx \phi, \\ \pi \to \rho \approx \omega \ \textit{ iff } \ \rho \approx \omega, & \pi \to \rho \approx \phi \ \textit{ iff } \ \pi \approx \omega \textit{ and } \rho \approx \phi. \end{array}$$

For every type π there is a minimum $n \in \mathtt{Nat}$, and types $\pi_1^1, \dots, \pi_1^n$, $\pi_2^1, \dots, \pi_2^n$ such that $\pi \approx \bigwedge_{i \leq n} \pi_1^i \to \pi_2^i$. Moreover, we have

$$\begin{array}{c} \bigwedge_{i \leq n} \pi_1^i \to \pi_2^i \preceq \bigwedge_{j \leq m} \rho_1^j \to \rho_2^j \textit{ iff} \\ \rho_2^j \not\approx \omega \textit{ implies } \exists J \subseteq \{1, ..., n\}, \bigwedge_{i \in J} \pi_1^i \succeq \rho_1^j, \bigwedge_{i \in J} \pi_2^i \preceq \rho_2^j \ (1 \leq j \leq m) \end{array} \tag{8}$$

Definition 28 extends to our setting the *aggregation* operation $\odot$ of [BEM09]. That operation has been introduced to interpret a parallel constructor corresponding to the $+$ of $\Lambda^{\|}$. Indeed, our setting allows to prove that the $\odot$ of [BEM09] is the sup of the preorder $\preceq$ (Proposition 29).

Definition 28. *We define the following operation over types by induction on their lengths:*

$$\begin{array}{ll} \phi \odot \pi := \pi & (\pi_1 \to \pi_2) \odot (\rho_1 \to \rho_2) := (\pi_1 \wedge \rho_2) \to (\pi_2 \odot \rho_2) \\ \omega \odot \pi := \omega & (\pi_1 \wedge \pi_2) \odot \rho := (\pi_1 \odot \rho) \wedge (\pi_2 \odot \rho) \\ (\pi_1 \to \pi_2) \odot \rho := \rho \odot (\pi_1 \to \pi_2) & \textit{if } \rho \textit{ is not an arrow} \end{array}$$

Notice that $\odot$ is associative and commutative modulo the associativity and commutativity of $\wedge$, in fact, (assuming π_i, ρ_i arrows)

$$(\pi_1 \wedge \pi_2) \odot (\rho_1 \wedge \rho_2) = ((\rho_1 \odot \pi_1) \wedge (\rho_2 \odot \pi_1)) \wedge ((\rho_1 \odot \pi_2) \wedge (\rho_2 \odot \pi_2))$$
$$\approx ((\pi_1 \odot \rho_1) \wedge (\pi_2 \odot \rho_2)) \wedge ((\pi_1 \odot \rho_2) \wedge (\pi_2 \odot \rho_2)) = (\rho_1 \wedge \rho_2) \odot (\pi_1 \wedge \pi_2)$$

Besides, $\odot$ is compatible with $\approx$, i.e. $\pi \approx \pi'$ and $\rho \approx \rho'$ entails $\pi \odot \rho \approx \pi' \odot \rho'$.

Proposition 29. *For any two types π, ρ, we have:*

$$\pi \wedge \rho \approx \inf(\pi, \rho) \qquad\qquad \pi \odot \rho \approx \sup(\pi, \rho)$$

Also, if π, ρ are not $\approx$-equivalent to ω (resp. ϕ) then so is $\pi \odot \rho$ (resp. $\pi \wedge \rho$).

Proof. By the characterization of the $\approx$-class of ω (resp. ϕ) given in Proposition 27 one can easily deduce that $\pi, \rho \not\approx \omega$ (resp. $\not\approx \phi$) entails $\pi \odot \rho \not\approx \omega$ (resp. $\pi \wedge \rho \not\approx \phi$).

That intersection is the inf is an immediate consequence of the rules of Figure 6(a). As for $\odot$ we proceed by structural induction on π and ρ.

The cases where one between π, ρ is equal to ϕ or ω are immediate consequences of the fact that the two constants are respectively the minimum and the maximum of $\preceq$ (Proposition 27).

If $\pi = \bigwedge_{i \leq n}(\pi_i^1 \to \pi_i^2)$ for an $n > 0$, then we have $\pi \odot \rho \approx \bigwedge_{i \leq n}\big((\pi_i^1 \to \pi_i^2) \odot \rho\big)$. Hence by induction hypothesis $\pi \odot \rho \approx \bigwedge_{i \leq n}\big(\sup(\pi_i^1 \to \pi_i^2, \rho)\big)$. This clearly implies $\pi \odot \rho \succeq \pi, \rho$. Let now $\zeta \succeq \pi, \rho$, we claim $\zeta \succeq \pi \odot \rho$. By Proposition 27, there is a minimum m such that $\zeta \approx \bigwedge_{i \leq m}(\zeta_j^1 \to \zeta_j^2)$. Note that the minimality condition implies either $\zeta \approx \omega$ or $\zeta_j^2 \not\approx \omega$, for all $j \leq m$. Since $\pi \preceq \zeta$, for every $j \leq m$, there is $J \subseteq \{1, \ldots, n\}$ s.t. $\bigwedge_{i \in J}(\pi_i^1 \to \pi_i^2) \preceq \zeta_j^1 \to \zeta_j^2$. Since also $\rho \preceq \zeta$, we have:

$$\bigwedge_{i \leq n}\big(\sup(\pi_i^1 \to \pi_i^2, \rho)\big) \preceq \bigwedge_{i \in J}\big(\sup(\pi_i^1 \to \pi_i^2, \rho)\big) \preceq \sup\big(\bigwedge_{i \in J}(\pi_i^1 \to \pi_i^2), \rho\big) \preceq \zeta_j^1 \to \zeta_j^2$$

The above inequality holds for every $j \leq m$, hence $\pi \odot \rho \preceq \zeta$.

If $\pi = \pi^1 \to \pi^2$, then the definition of $\pi \odot \rho$ depends on ρ. The cases $\rho = \phi, \omega$ have been already considered and $\rho = \rho^1 \wedge \rho^2$ turns to the previous one. So, let $\rho = \rho^1 \to \rho^2$ and $\pi \odot \rho = (\pi^1 \wedge \rho^1) \to (\pi^2 \odot \rho^2)$. By induction hypothesis this gives, $\pi \odot \rho = \inf(\pi^1, \rho^1) \to \sup(\pi^2, \rho^2)$, which immediately implies $\pi, \rho \preceq \pi \odot \rho$. The proof that for every $\zeta \succeq \pi, \rho$, we have $\zeta \succeq \pi \odot \rho$ is an easy variant of the case π is a proper intersection. $\square$

The next lemma is an easy variant in the setting of $\Lambda^{\parallel}$ of the Generation Lemma in [RDRP04] (Lemma 10.1.7).

Lemma 30 (Generation). *1. $\Gamma \vdash_{\mathtt{M}} x : \pi$ implies $\Gamma(x) \preceq \pi$.*

2. $\Gamma \vdash_{\mathtt{M}} \lambda x.M : \pi$ implies either $\pi \approx \omega$ or there are $\pi_1, \ldots, \pi_n$ such that $\pi_1 \wedge \ldots \wedge \pi_n \preceq \pi$, $\pi_i = \zeta_i \to \rho_i$ and $\Gamma, x : \zeta_i \vdash_{\mathtt{M}} M : \rho_i$ for every $i \in \{1, \ldots, n\}, n > 1$.

3. $\Gamma \vdash_{\mathtt{M}} \lambda x.M : \pi \to \rho$ implies $\Gamma, x : \pi \vdash_{\mathtt{M}} M : \rho$.

4. $\Gamma \vdash_{\mathtt{M}} MP : \pi$ *implies there are* $\pi_1, \ldots, \pi_n$*, for* $n \geq 1$*, s.t.* $\pi_1 \wedge \ldots \wedge \pi_n \preceq \pi$ *and for all* $1 \leq i \leq n$*,* $\Gamma \vdash_{\mathtt{M}} M : \rho_i \to \pi_i$ *and* $\Gamma \vdash_{\mathtt{M}} P : \rho_i$*.*

5. $\Gamma \vdash_{\mathtt{M}} [M^!]{\cdot}P : \pi$ *implies* $\Gamma \vdash_{\mathtt{M}} M : \pi$ *and, if* P *is non-empty, also* $\Gamma \vdash_{\mathtt{M}} P : \pi$*.*

6. $\Gamma \vdash_{\mathtt{M}} \mathbb{M} : \pi$ *implies that, for every term* M *of the sum* $\mathbb{M}$*,* $\Gamma \vdash_{\mathtt{M}} M : \pi$*.*

7. $\Gamma, x : \pi \vdash_{\mathtt{M}} \mathbb{A} : \rho$ *and* $x \notin \mathrm{FV}(\mathbb{A})$ *imply* $\Gamma \vdash_{\mathtt{M}} \mathbb{A} : \rho$*.*

In what follows we prove that the set of types assigned by $\vdash_{\mathtt{M}}$ is invariant under middle (hence giant) reduction (Proposition 34).

Lemma 31 (Substitution)**.** *If* $\Gamma, x : \pi \vdash_{\mathtt{M}} \mathbb{A} : \rho$ *and* $\Gamma \vdash_{\mathtt{M}} N : \pi$*, then there are* $\Gamma, x : \pi \vdash_{\mathtt{M}} A\{(N + x)/x\} : \rho$ *and* $\Gamma \vdash_{\mathtt{M}} A\{N/x\} : \rho$*.*

Proof. By structural induction on $\mathbb{A}$. If $\mathbb{A} = x$, Lemma 30.1 states $\pi \preceq \rho$. Then a derivation of $\Gamma, x : \pi \vdash_{\mathtt{M}} A\{(N + x)/x\} : \rho$ can be obtained by applying rule $\preceq$ to the derivation of $\Gamma \vdash_{\mathtt{M}} N : \pi$. Moreover, we can build the derivation:

$$\dfrac{\dfrac{\Gamma, x : \pi \vdash_{\mathtt{M}} x : \pi}{\Gamma, x : \pi \vdash_{\mathtt{M}} x : \rho}\,\preceq \qquad \dfrac{\dfrac{\dfrac{}{\Gamma \vdash_{\mathtt{M}} N : \pi}}{\Gamma, x : \pi \vdash_{\mathtt{M}} N : \pi}\,\mathtt{weak}}{\Gamma, x : \pi \vdash_{\mathtt{M}} N : \rho}\,\preceq}{\Gamma, x : \pi \vdash_{\mathtt{M}} N + x : \rho}\,\mathtt{mix+}$$

If $\mathbb{A} = [M^!]{\cdot}P$, we consider P non-empty, and we build only $\Gamma, x : \pi \vdash_{\mathtt{M}} [M^!]{\cdot} P\{(N + x)/x\} : \rho$: the case P empty and the definition of $\Gamma \vdash_{\mathtt{M}} A\{N/x\} : \rho$ are easy variants. So, let $\mathbb{A}\{N + x/x\} = [M_1^!, \ldots, M_k^!]{\cdot}P\{N + x/x\}$, where $M\{N + x/x\} = \sum_{i=0}^{k} M_i$ and $P\{N + x/x\}$ is always a simple bag, P not containing linear resources. By Lemma 30.5, we have two derivations of $\Gamma, x : \pi \vdash_{\mathtt{M}} [M^!] : \rho$ and $\Gamma, x : \pi \vdash_{\mathtt{M}} P : \rho$. The induction hypothesis yields $\Gamma, x : \pi \vdash_{\mathtt{M}} [M_1^!, \ldots, M_k^!] : \rho$ and $\Gamma, x : \pi \vdash_{\mathtt{M}} P\{N + x/x\} : \rho$. By Lemma 30, $\Gamma \vdash_{\mathtt{M}} M_i : \rho$, and the result follows by applying n times the rule $\mathtt{mix}\cdot$.

The other cases are straightforward or easy variants of the previous one. $\square$

Similarly to above, the following lemmas are proven by easy structural inductions on $\mathbb{A}$.

Lemma 32 (Partial expansion)**.** *Let* $\Gamma, x : \pi \vdash_{\mathtt{M}} \mathbb{A}\{N + x/x\} : \rho$*, then there are is a type* $\zeta \succeq \pi$ *such that* $\Gamma, x : \zeta \vdash_{\mathtt{M}} \mathbb{A} : \rho$ *and* $\Gamma \vdash_{\mathtt{M}} N : \zeta$*.*

Lemma 33 (Total expansion)**.** *Let* $x \notin d(\Gamma)$*, if* $\Gamma \vdash_{\mathtt{M}} \mathbb{A}\{N/x\} : \rho$ *then there is a type* π *such that* $\Gamma, x : \pi \vdash_{\mathtt{M}} \mathbb{A} : \rho$ *and* $\Gamma \vdash_{\mathtt{M}} N : \pi$*.*

Proposition 34 (Invariance of $\vdash_{\mathtt{M}}$ typings)**.** *Let* $\epsilon \in \{\mathtt{m}, \mathtt{g}\}$ *and* $M \xrightarrow{\epsilon} \mathbb{M}$*, then* M *and* $\mathbb{M}$ *share the same judgements, i.e.* $\Gamma \vdash_{\mathtt{M}} M : \pi$ *iff* $\Gamma \vdash_{\mathtt{M}} \mathbb{M} : \pi$*.*

Proof. The proof is by induction on the context enclosing the redex reduced in $M \xrightarrow{\epsilon} \mathbb{M}$. The induction steps are immediate, while the base of induction is when M is the redex fired by the reduction $M \xrightarrow{\epsilon} \mathbb{M}$. One can consider only the middle-step cases, the giant one will follow since it corresponds to a sequence of middle-steps.

Suppose $M = (\lambda x.L)[N^!]{\cdot}P$, we suppose also P non-empty, the case P is empty being simpler (using Lemma 33 instead of Lemma 32). So, let $M \xrightarrow{\mathtt{m}} (\lambda x.L\{N + x/x\})P = \mathbb{M}$ and assume $\Gamma \vdash_{\mathtt{M}} (\lambda x.L)[N^!]{\cdot}P : \pi$. Generation (Lemma 30) allows us to say that $\pi \approx \pi_1 \wedge \ldots \wedge \pi_n$, and, for all i, there is ρ_i such that $\Gamma, x : \rho_i \vdash_{\mathtt{M}} L : \pi_i$ and $\Gamma \vdash_{\mathtt{M}} N : \rho_i$, and $\Gamma \vdash_{\mathtt{M}} P : \rho_i$. By Lemma 31 and the first two judgements, we get a derivation $\Phi_i :: \Gamma, x : \rho_i \vdash_{\mathtt{M}} L\{N + x/x\} : \pi_i$. For each $i \leq n$, we get a derivation of $\Psi_i :: \Gamma \vdash_{\mathtt{M}} (\lambda x.L\{N + x/x\})P : \pi_i$ by applying to Φ_i one $\to\mathtt{I}$ rule and one $\to\mathtt{E}$ rule with right premise $\Gamma \vdash_{\mathtt{M}} P : \rho_i$. Then, gathering all Ψ_i's with $n-1$ rules $\wedge I$ and applying one $\preceq$ yields a derivation with conclusion $\Gamma \vdash_{\mathtt{M}} (\lambda x.L\{N + x/x\})P : \pi$.

For the converse, let $\Gamma \vdash_{\mathtt{M}} (\lambda x.L\{N + x/x\})P = \mathbb{M} : \pi$. Having all bags just reusable resources, the variable x can have at most one linear occurrence in L (and an arbitrary number of reusable ones). $\mathbb{M}$ is either a simple term or a sum having two terms. Let us consider the last case, the former being an easier variant. So, we have $L\{N + x/x\} = L^1 + L^2$ and $\mathbb{M} = (\lambda x.L^1)P + (\lambda x.L^2)P$. By applying Generation, we get, for $i = 1, 2$, $\Gamma \vdash_{\mathtt{M}} (\lambda x.L^i)P : \pi$ and then $\pi^i_1 \wedge \ldots \wedge \pi^i_{n_i} \preceq \pi$ and $\rho^i_1, \ldots, \rho^i_{n_i}$ such that for every $j \leq n_i$, $\Gamma \vdash_{\mathtt{M}} \lambda x.L^i : \rho^i_j \to \pi^i_j$ and $\Gamma \vdash_{\mathtt{M}} P : \rho^i_j$. Notice that for every i, j, $\bigwedge_{i,j} \rho^i_j \preceq \rho^i_j$, so $\rho^i_j \to \pi^i_j \preceq (\bigwedge_{i,j} \rho^i_j) \to \pi^i_j$, and so by a $\preceq$-rule we have $\Gamma \vdash_{\mathtt{M}} \lambda x.L^i : (\bigwedge_{i,j} \rho^i_j) \to \pi^i_j$. Thus, a number of $\wedge I$-rules gives us $\Gamma \vdash_{\mathtt{M}} \lambda x.L^i : \bigwedge_{i,j}\big((\bigwedge_{i,j} \rho^i_j) \to \pi^i_j\big)$ and, since $\bigwedge_{i,j}\big((\bigwedge_{i,j} \rho^i_j) \to \pi^i_j\big) \preceq (\bigwedge_{i,j} \rho^i_j) \to (\bigwedge_{i,j} \pi^i_j) \preceq (\bigwedge_{i,j} \rho^i_j) \to \pi$, we have $\Gamma \vdash_{\mathtt{M}} \lambda x.L^i : (\bigwedge_{i,j} \rho^i_j) \to \pi$. By Generation (Lemma 30), $\Gamma, x : \bigwedge_{i,j} \rho^i_j \vdash_{\mathtt{M}} L^i : \pi$ and by a $\mathtt{mix}+$-rule $\Gamma, x : \bigwedge_{i,j} \rho^i_j \vdash_{\mathtt{M}} L\{N + x/x\} : \pi$. We can now apply Lemma 32, getting a type $\zeta \succeq \bigwedge_{i,j} \rho^i_j$ and derivations $\Phi :: \Gamma, x : \zeta \vdash_{\mathtt{M}} L : \pi$ and $\Psi :: \Gamma \vdash_{\mathtt{M}} N : \zeta$. Since $\Gamma \vdash_{\mathtt{M}} P : \bigwedge_{i,j} \rho^i_j$, by one $\preceq$-rule we have $\Gamma \vdash_{\mathtt{M}} P : \zeta$, hence, by a $\mathtt{mix}\cdot$-rule $\Gamma \vdash_{\mathtt{M}} P \cdot [N^!] : \zeta$. We conclude by one rule $\to\mathtt{I}_n$ and one $\to\mathtt{E}$. $\square$

Lemma 35. *Every must-outer-normal form of $\Lambda^{\|}$ is typable in $\vdash_{\mathtt{M}}$ with a non-trivial type, i.e. a type $\not\approx \omega$.*

Proof. We do induction on the number $n \geq 1$ of terms in a Monf $\mathbb{M}$. Let $n = 1$, then $\mathbb{M}$ has the following shape: $M = \lambda x_1 \ldots x_s.yP_1...P_p$. Let $y = x_j$ (the case y is free being an easy variant), for some $j \leq s$. Let $\Gamma(x_k) = \omega$, for every $k \neq j$ and let $\Gamma(x_j) = \underbrace{\omega \to \ldots \to \omega}_{p \text{ times}} \to \phi$. Recall that, for all $i \leq p$, $\Gamma \vdash_{\mathtt{M}} P_i : \omega$, hence by $p-1$ rules $\to\mathtt{E}$ we obtain a derivation of $\Gamma \vdash_{\mathtt{M}} x_j P_1..P_p : \phi$, and then by s rules $\to\mathtt{I}$ we get $\vdash_{\mathtt{M}} \lambda x_1...x_s.x_j P_1..P_p : \pi$, where

$$\pi = \underbrace{\omega \to \ldots \to \omega}_{j-1} \to (\underbrace{\omega \to \ldots \to \omega}_{p} \to \phi) \to \underbrace{\omega \to \ldots \to \omega}_{s-j} \to \phi,$$

which is a type $\not\approx \omega$ by Lemma 27.

As for the induction case, let $\mathbb{M} = \mathbb{M}' + \mathbb{M}''$, with both $\mathbb{M}', \mathbb{M}''$ different from 0. By the definition of Monf, both $\mathbb{M}', \mathbb{M}''$ are Monf and hence by the induction hypothesis $\Gamma' \vdash_{\mathtt{M}} \mathbb{M}' : \sigma'$ and $\Gamma'' \vdash_{\mathtt{M}} \mathbb{M}'' : \sigma''$, with $\sigma', \sigma'' \not\approx \omega$.

Define the following context having domain equal to $d(\Gamma') \cup d(\Gamma'')$:

$$\Gamma(x) = \begin{cases} \pi & \text{if } x \notin d(\Gamma') \cap d(\Gamma'') \text{ and } \Gamma'(x) \text{ or } \Gamma''(x) \text{ is } \pi, \\ \pi' \wedge \pi' & \text{if } \Gamma'(x) = \pi' \text{ and } \Gamma''(x) = \pi'' \end{cases}$$

Since the rules $\mathtt{weak}$ and $\preceq \mathtt{L}$ are admissible, we get $\Gamma \vdash_{\mathtt{M}} \mathbb{M}' : \sigma'$ and $\Gamma \vdash_{\mathtt{M}} \mathbb{M}'' : \sigma''$. Then, since $\sigma' \odot \sigma'' \succeq \sigma', \sigma''$ (Proposition 29), we have $\Gamma \vdash_{\mathtt{M}} \mathbb{M}' : \sigma' \odot \sigma''$ and $\Gamma \vdash_{\mathtt{M}} \mathbb{M}'' : \sigma' \odot \sigma''$, and so $\Gamma \vdash_{\mathtt{M}} \mathbb{M} : \sigma' \odot \sigma''$ by one $\mathtt{mix}+$-rule. We have $\sigma' \odot \sigma'' \not\approx \omega$ still by Proposition 29. $\square$

In order to prove that having a non-trivial type in the system $\vdash_{\mathtt{M}}$ implies to be must-outer normalizable by means of an outer reduction sequence, we will use a computability argument. The proof is an adaptation of the proof of approximation in the model D_∞, given in [PRDR04], and it is given in the next subsection.

Theorem 36. *Given a term $M \in \Lambda^{\|}$, the following are equivalent:*

1. *M is must-outer normalizable,*
2. *M is non-trivially typable by $\vdash_{\mathtt{M}}$,*
3. *M is reducible to a* Monf *by outer ϵ reduction, $\epsilon \in \{\mathtt{m}, \mathtt{g}\}$,*
4. *M is must-solvable in $\Lambda^{\|}$.*

Proof. $1 \Rightarrow 2$: by Lemma 35 and Proposition 34. $2 \Rightarrow 3$: by Theorem 42. $3 \Rightarrow 4$: by Theorem 13. $4 \Rightarrow 1$: if M is must-solvable then there is a outer-context $C(\cdot)$ such that $C(M) \xrightarrow{\mathtt{g}*} n\mathbf{I}$. Since the latter is a Monf, we have for the already proven $1 \Rightarrow 2$ that $C(M)$ is typable with a type $\rho \not\approx \omega$. Since $C(\cdot)$ is a outer-context, then M is typable with a type $\pi \not\approx \omega$ (this can be easily deduced by inspecting the rules of Figure 6(b) and the characterization of the ω $\approx$-class in Lemma 27). Then, by Theorem 42 M has a Monf. $\square$

5.2 Non-trivial typability entails the finiteness of $\xrightarrow{\circ}$.

We adapt to our setting a variant of the saturated sets technique due to [Kri93]. In this section, outer reduction is short for both middle and giant outer reduction, and $\xrightarrow{\circ}$ denotes $\xrightarrow{\circ\epsilon}$, with $\epsilon \in \{\mathtt{m}, \mathtt{g}\}$.

Definition 37. *We interpret types as sets of resource terms as follows:*

$$[\![\phi]\!] := \{\mathbb{M} \mid \mathbb{M} \xrightarrow{\circ*} \sum_i x_i P_{i,1} \dots P_{i,p_i} \mid i \leq n > 0, p_i \geq 0, P_{i,j} \in \Lambda^{b\|}\};$$

$[\![\omega]\!] := \mathtt{Nat}^+\langle\Lambda^{\|}\rangle$; $[\![\pi \to \rho]\!] := \{\mathbb{M} \mid \forall \mathbb{N} \in [\![\pi]\!], \mathbb{M}[\mathbb{N}^!] \in [\![\rho]\!]\}$; $[\![\pi \wedge \rho]\!] := [\![\pi]\!] \cap [\![\rho]\!]$.

The following three lemmas show that this interpretation respects the preorder on types, assures the outer normalization for non-trivial types and is closed under outer expansion.

Lemma 38 (Preorder)**.** *For any types π and ρ, $\pi \preceq \rho$ entails $[\![\pi]\!] \subseteq [\![\rho]\!]$. In particular, $[\![\phi]\!] \subseteq [\![\pi]\!]$ and for any variable x, $x \in [\![\pi]\!]$.*

Proof. By induction on π, considering all cases defining $\preceq$ (Figure 6(a)). Then, Lemma 27 allows to deduce $[\![\phi]\!] \subseteq [\![\pi]\!]$ and $x \in [\![\pi]\!]$, since $x \in [\![\phi]\!]$ by definition.

The only delicate case is $\phi \preceq \omega \to \phi$. Let $\mathbb{M} \in [\![\phi]\!]$, i.e. $\mathbb{M} \xrightarrow{\circ*} \sum_i x^i P_1^i \dots P_{n_i}^i$. We prove that for any $\mathbb{N} \in \mathtt{Nat}^+\langle\Lambda^{\|}\rangle$, $\mathbb{M}[\mathbb{N}^!] \in [\![\phi]\!]$. Indeed, any term of a sum that is an outer reduct of $\mathbb{M}$ cannot be a λ abstraction, since otherwise $\mathbb{M} \not\xrightarrow{\circ*} \sum_i x^i P_1^i \dots P_{n_i}^i$. Hence, $\mathbb{M}[\mathbb{N}^!] \xrightarrow{\circ*} \sum_i x^i P_1^i \dots P_{n_i}^i[\mathbb{N}^!]$, i.e. $\mathbb{M}[\mathbb{N}^!] \in [\![\phi]\!]$. $\square$

Lemma 39 (Outer normalization)**.** *For every type $\pi \not\approx \omega$, for every $\mathbb{M} \in [\![\pi]\!]$, $\mathbb{M}$ is reducible to a must-outer-normal form by the outer reduction.*

Proof. By induction on π. The case $\pi = \phi$ is by definition of $[\![\phi]\!]$. If $\pi = \pi_1 \wedge \pi_2$ then by Lemma 27 there is $\pi_i \not\approx \omega$. By induction hypothesis the terms in $[\![\pi_i]\!]$ are outer normalizable and we conclude by $[\![\pi]\!] \subseteq [\![\pi_i]\!]$. If $\pi = \pi_1 \to \pi_2$, then let $\mathbb{M} \in [\![\pi_1 \to \pi_2]\!]$. By Lemma 38 any variable $x \in [\![\pi_1]\!]$, so $\mathbb{M}[x^!] \in [\![\pi_2]\!]$. By Lemma 27, $\pi_2 \not\approx \omega$, so by induction $\mathbb{M}[x^!]$ is outer normalizable. We conclude by the simple remark that if $\mathbb{M}[x^!]$ is outer normalizable then so is $\mathbb{M}$. □

Lemma 40 (Saturation)**.** *For every type π, if $\mathbb{M}\,\{\mathbb{N}/x\}\,P_1 \dots P_n \in [\![\pi]\!]$, we have $(\lambda x.\mathbb{M})[\mathbb{N}^!]P_1 \dots P_n \in [\![\pi]\!]$.*

Proof. By induction on π. The case $\pi = \omega, \phi$ are straightforward. If $\pi = \pi_1 \to \pi_2$, suppose $\mathbb{M}\,\{\mathbb{N}/x\}\,P_1 \dots P_n \in [\![\pi_1 \to \pi_2]\!]$, and let $\mathbb{L} \in [\![\pi_1]\!]$. We have $\mathbb{M}\,\{\mathbb{N}/x\}\,P_1 \dots P_n[\mathbb{L}^!] \in [\![\pi_2]\!]$ and, by induction on π_2, $(\lambda x.\mathbb{M})[\mathbb{N}^!]P_1 \dots P_n[\mathbb{L}^!] \in [\![\pi_2]\!]$. We conclude $(\lambda x.\mathbb{M})[\mathbb{N}^!]P_1 \dots P_n \in [\![\pi_1 \to \pi_2]\!]$. The case $\pi = \pi_1 \wedge \pi_2$ is an immediate consequence of the induction hypothesis applied π_1 and π_2. □

The adequacy lemma relates the interpretation on types with the derivations.

Lemma 41 (Adequacy)**.** *Let $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} \mathbb{M} : \rho$ and for every $i \leq n$, $\mathbb{N}_i \in [\![\pi_i]\!]$, we have $\mathbb{M}\,\{\mathbb{N}_1/x_1\} \dots \{\mathbb{N}_n/x_n\} \in [\![\rho]\!]$.*

Proof. By structural induction on a derivation Φ of $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} \mathbb{M} : \rho$. The case `var` and ω are immediate. If Φ ends in a $\to$`I` rule with premise $x_1 : \pi_1, \dots, x_n : \pi_n, y : \zeta \vdash_{\mathtt{M}} M' : \rho$ (where $\mathbb{M} = \lambda y.M'$), then by induction we have, for any $i \leq n$, $\mathbb{N}_i \in [\![\pi_i]\!]$ and $\mathbb{L} \in [\![\zeta]\!]$, $M'\,\{\mathbb{N}_1/x_1\} \dots \{\mathbb{N}_n/x_n\}\,\{\mathbb{L}/y\} \in [\![\rho]\!]$. By Lemma 40, we have $(\lambda y.M'\,\{\mathbb{N}_1/x_1\} \dots \{\mathbb{N}_n/x_n\})[\mathbb{L}^!] \in [\![\rho]\!]$, hence $(\lambda y.M')\,\{\mathbb{N}_1/x_1\} \dots \{\mathbb{N}_n/x_n\} \in [\![\zeta \to \rho]\!]$.

If Φ ends in a $\to$`E` rule with premises $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} M' : \zeta \to \rho$ and $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} P : \zeta$ (where $\mathbb{M} = M'P$), then one concludes easily by induction hypothesis once remarked that Φ must have a subproof of $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} N : \zeta$ for every term N in the bag P.

The case $\wedge I$ is an immediate consequence of the induction hypothesis and the case of a $\preceq$ rule is a consequence of the induction and of Lemma 38. □

Theorem 42. *For any term $M \in \Lambda^{\|}$ and for any type $\pi \not\approx \omega$, if $\Gamma \vdash_{\mathtt{M}} M : \pi$, then M is reducible to a must-outer-normal form by outer reduction.*

Proof. Let $x_1 : \pi_1, \dots, x_n : \pi_n \vdash_{\mathtt{M}} M : \pi$, by Lemma 38, $x_i \in [\![\pi_i]\!]$, then by Lemma 41, $M = M\,\{x_1/x_1\} \dots \{x_n/x_n\} \in [\![\pi]\!]$. We conclude by Lemma 39. □

6 Conclusion and future work

We did study the notions of may and must-solvability in the resource calculus. We succeeded in characterizing completely the may-solvability, from the syntactical, operational and logical point of view. The notion of must solvability has been completely characterized only for the purely non deterministic fragment. In fact, the problem of characterizing the must-solvability in the whole calculus seems to coincide with the problem of semantically separating the two notions of failure and non termination, which is an open problem, for the moment.

The two type assignment systems we used for the logical characterization of may-solvability in Λ^r and of must-solvability in $\Lambda^{\|}$ can be the starting point for providing a denotational semantics in logical forms of the two calculi. In particular $\vdash_{\mathtt{m}}$ can be presented as a logical description of a denotational model for the resource calculus, where all the may-unsolvable terms are equated. Indeed such a goal seems to us non immediate, since a quantitative account of resources does not fit well with the contextual closure of the interpretation function. For a discussion about this point see [dC09]. A possible solution might be achieved following the ideas in [BEM07].

References

[Bar84] Henk Barendregt. *The Lambda-Calculus, its Syntax and Semantics*. Stud. Logic Found. Math., vol. 103. North-Holland, 1984.

[BCL99] Gerard Boudol, Pierre-Louis Curien, and Carolina Lavatelli. A Semantics for Lambda Calculi with Resources. *MSCS*, 9(5):437–482, 1999.

[BEM07] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not Enough Points Is Enough. In *CSL*, volume 4646 of *Lecture Notes in Comp. Sci.*, pages 298–312, 2007.

[BEM09] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Mazonetto. A relational semantics for parallelism and non-determinism in a functional setting. Preprint, 2009.

[Bou93] Gérard Boudol. The Lambda-Calculus with Multiplicities. *INRIA Report 2025*, 1993.

[CDCV80] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Principal Type Schemes and Lambda-Calculus Semantics. In *To H. B. Curry. Essays on Combinatory Logic, Lambda-calculus and Formalism*, pages 480–490. Accademic Press, 1980.

[CDCV81] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri. Functional Characters of Solvable Terms. *Zeitschrift für Mathematische Logik*, 27:45–58, 1981.

[dC09] Daniel de Carvalho. Execution Time of λ-Terms via Denotational Semantics and Intersection Types. Preprint, 2009.

[dCPTdF08] Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A Semantic Measure of the Execution Time in Linear Logic. *Th. Comp. Sc.*, 2008. to appear.

[dP95] Ugo de'Liguoro and Adolfo Piperno. Non Deterministic Extensions of Untyped Lambda-Calculus. *Inf. Comput.*, 122(2):149–177, 1995.

[ER03] Thomas Ehrhard and Laurent Regnier. The Differential Lambda-Calculus. *Theor. Comput. Sci.*, 309(1):1–41, 2003.

[ER06] Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms. In *CiE*, volume 3988 of *LNCS*, pages 186–197, 2006.

[ER08] Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor Expansion of Ordinary Lambda-Terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.

[Hyl76] J. Martin E. Hyland. A Syntactic Characterization of the Equality in Some Models of the Lambda Calculus. *J. London Math. Soc.*, 2(12):361–370, 1976.

[Kfo00] Assaf J. Kfoury. A Linearization of the Lambda-Calculus and Consequences. *J. Logic Comp.*, 10(3):411–436, 2000.

[Kri93] Jean-Louis Krivine. *Lambda-calculus, types and models.* Ellis Horwood, 1993.

[NM04] Peter M. Neergaard and Harry G. Mairson. Types, Potency, and Idempotency: why Nonlinearity and Amnesia Make a Type System Work. In Okasaki and Fisher, editors, *9th ACM SIGPLAN ICFP 2004*, pages 138–149. ACM, 2004.

[PRDR04] Luca Paolini and Simona Ronchi Della Rocca. The parametric parameter passing λ-calculus. *Information and Computation*, 189(1):87–106, February 2004.

[PRDR10] Michele Pagani and Simona Ronchi Della Rocca. Solvability in Resource Lambda-Calculus. In Luke Ong, editor, *FOSSACS*, volume 6014 of *Lecture Notes in Comp. Sci.*, pages 358–373, 2010.

[PT09] Michele Pagani and Paolo Tranquilli. Parallel Reduction in Resource Lambda-Calculus. In *APLAS*, volume 5904 of *LNCS*, pages 226–242, 2009.

[RDRP04] Simona Ronchi Della Rocca and Luca Paolini. *The Parametric λ-Calculus: a Metamodel for Computation.* EATCS Series. Springer, Berlin, 2004.

[Sco76] Dana S. Scott. Data types as lattices. *SIAM J. Comput.*, 5(3):522–587, 1976.

[Tra08] Paolo Tranquilli. Intuitionistic Differential Nets and Lambda-Calculus. *Theor. Comput. Sci.*, 2008. to appear.

[Tra09] Paolo Tranquilli. *Nets between Determinism and Nondeterminism.* Ph.D. thesis, Università Roma Tre/Université Paris Diderot (Paris 7), April 2009.

[Val01] Silvio Valentini. An elementary proof of strong normalization for intersection types. *Archive for Mathematical Logic*, 40(7):475–488, October 2001.

[Vau09] Lionel Vaux. The algebraic lambda calculus. *Math. Struct. Comp. Sci.*, 19(5):1029–1059, 2009.

[WDMT02] J. B. Wells, Allyn Dimock, Robert Muller, and Franklyn Turbak. A Calculus with Polymorphic and Polyvariant Flow Types. *J. Funct. Program.*, 12(3):183–227, 2002.