Solvability in Resource Lambda Calculus

Solvability in Resource Lambda-Calculus*

Michele Pagani and Simona Ronchi della Rocca

Dipartimento di Informatica – Università di Torino
C.so Svizzera 185 – 10149 Torino (IT)
{pagani,ronchi}@di.unito.it

Abstract. The resource calculus is an extension of the λ-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, giving rise to non-deterministic choices, expressed by a formal sum. Using the λ-calculus terminology, we call *solvable* a term that can interact with the environment: solvable terms represent meaningful programs. Because of the non-determinism, different definitions of solvability are possible in the resource calculus. Here we study the optimistic (angelical, or may) notion, and so we define a term *solvable* whenever there is a simple head context reducing the term into a sum where at least one addend is the identity. We give a syntactical, operational and logical characterization of this kind of solvability.

1 Introduction

The resource calculus (Λ^r) is an extension of the λ-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. In this setting 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 no internal choice is performed actually, the result being a formal sum of all the possible cases. In case of a multiset of linear resources, also a notion of *crash* arises, whenever the cardinality of the multiset does not fit exactly the number of occurrences. Then 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 [1]. Ehrhard and Regnier designed the differential λ-calculus [2], 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, which defined the Λ^r syntax, and

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

showed a Curry-Howard correspondence between this calculus and Ehrhard and Regnier's differential nets [3]. 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. Clearly, 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 [4]. On the other side, allowing only multisets of linear terms gives the linear fragment of Λ^r, used by Ehrhard and Regnier for giving a quantitative account to λ-calculus β-reduction through Taylor expansion [5, 6].

The aim of this paper is to study the operational behaviour of the full resource calculus. It has been already proved that it enjoys the properties of confluence and a sort of standardization [7]. In particular confluence does not clash with non-determinism since the sum carries all the possibilities. Here we study the solvability property. Namely, following the λ-calculus terminology, we use the word *solvable* in order to denote 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.

Let us recall that in the λ-calculus a term M is defined to be solvable if and only if there is a context $C(\cdot)$ (of a non constant behaviour) such that $C(M)$ 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 [8], operationally if and only if the head reduction strategy applied to it eventually stops [8], logically if and only if it can be typed in a suitable intersection type assignment system [9], denotationally if and only if its denotation is not minimal in a suitable sensible model [10, 11]. Our aim is to characterize the notion of solvability in Λ^r following the same lines.

The first problem we meet is the definition of solvability in Λ^r. In this paper we decided to follow an optimistic (angelical, or may) approach, and so we define a term to be solvable whenever there is a context, of a non constant behaviour, that, when filled by the term, reduces to a sum of terms, at least one of these being the identity. Other possible definitions of solvability (on which we are currently working) are discussed in the conclusion of the paper.

Our result is a characterization of solvability in Λ^r from a syntactical, operational and logical point of view (Theorem 19). It turns out that an extended notion of head-normal form can be defined, 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 [7], where no reduction is made inside reusable resources, and we prove that in order to reach the head-normal form we can restrict ourselves to use just reduction strategies of this kind. Moreover we give also a logical characterization of solvability, through a

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	$::= [M_1^{(!)}, \ldots, M_n^{(!)}]$	bags
$\Lambda^{(b)}$:	A, B	$::= M \mid P$	expressions
$\mathbb{M}, \mathbb{N} \in \mathcal{N}\langle\Lambda^r\rangle$	$\mathbb{P}, \mathbb{Q} \in \mathcal{N}\langle\Lambda^b\rangle$	$\mathbb{A}, \mathbb{B} \in \mathcal{N}\langle\Lambda^{(b)}\rangle := \mathcal{N}\langle\Lambda^r\rangle \cup \mathcal{N}\langle\Lambda^b\rangle$	sums

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

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

$$[(\textstyle\sum_i M_i)]\cdot P := \textstyle\sum_i [M]_i \cdot P \qquad [(\textstyle\sum_i M_i)^!]\cdot P := [M_1^!, \ldots, M_k^!]\cdot P$$

(b) Notation on $\mathcal{N}\langle\Lambda^{(b)}\rangle$.

Fig. 1: Syntax of resource calculus. The symbol $\mathcal{N}$ denotes the set of natural numbers, $\mathcal{N}\langle\Lambda^r\rangle$ (resp. $\mathcal{N}\langle\Lambda^b\rangle$) denotes the set of finite formal sums of terms (resp. bags), with 0 referring to the neutral element.

type assignment system, assigning to terms suitable non-idempotent intersection types. All these characterizations are conservative with respect to the λ-calculus.

The type assignment system we define 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 [12]. We plan to continue our investigation in the direction of giving a clear setting where our type assignment can be presented as a logical description of a denotational model for the resource calculus, where all the 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 [12]. A possible solution might be achieved following the ideas in [13].

The paper is organized as follows. Section 2 contains a syntactical description of the resource calculus. Section 3 is dedicated to the definition of solvability and of head-normal form. In Section 4 the intersection type assignment system is presented and its properties are stated. In Section 5 there is the proof of the main theorem, showing all the characterizations of solvability. In Section 6 alternative notions of solvability are discussed.

2 Resource Calculus

Syntax. 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

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

$$(MP)\langle N/x\rangle := M\langle N/x\rangle P + M(P\langle N/x\rangle),$$

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

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

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

with their typical metavariables. A resource can be linear (it must be used exactly once) or not (it can be used ad libitum), 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. It must be noted though that we will never omit the dot $\cdot$, to avoid confusion with application. 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. In writing $\mathcal{N}\langle\Lambda^{(b)}\rangle$ we are abusing the notation, as it does not denote the $\mathcal{N}$-module generated over $\Lambda^{(b)} = \Lambda^r \cup \Lambda^b$ but rather the union of the two $\mathcal{N}$-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 on the "surface". 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. Notice the similarity between the equation $[(M+N)^!] = [M^!]{\cdot}[N^!]$ and $\mathrm{e}^{x+y} = \mathrm{e}^x{\cdot}\mathrm{e}^y$: this is far from a coincidence, as Taylor expansion and linear logic semantics show well [6]. We adopt the usual λ-calculus conventions as in [8]. Also we use the 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. Due to the presence of two kinds of resources, we need two different notions of substitutions, so to capture both the linear and non linear behaviour. Moreover we define also a resource substitution, which is expressed in function of the first two, useful for defining the reduction.

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 $\mathbb{A}$[1] 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 $\mathbb{N}$.*

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

3. **Resource substitution** $A\langle\!\langle N^{(!)}/x\rangle\!\rangle$ *is the disjoint union of the partial and linear substitutions, i.e.* $A\langle\!\langle N/x\rangle\!\rangle := A\langle N/x\rangle$ *and* $A\langle\!\langle N^!/x\rangle\!\rangle := A\{N + x/x\}$.

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 $(y[x][x])\langle M/x\rangle = y[M][x] + y[x][M]$. Indeed linear substitution bears resemblance to differentiation, as it is in Ehrhard and Regnier's differential λ-calculus [2]. We refer to [3, 14] for the mathematical intuitions underlying the resource calculus. 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.

Lemma 2 ([2, 3, 14]). *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*

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

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

Furthermore we have, if $x \notin \mathrm{FV}(\mathbb{M}) \cup \mathrm{FV}(\mathbb{N})$,

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

Reductions. 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 $\mathcal{N}\langle\Lambda^r\rangle$. The expression $\mathbb{C}(M)$ denotes the result of blindly replacing M to the hole (allowing variable capture) in $C(\cdot)$. 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 r in $\Lambda^r \times \mathcal{N}\langle\Lambda^r\rangle$ is extended to one in $\mathcal{N}\langle\Lambda^r\rangle \times \mathcal{N}\langle\Lambda^r\rangle$ by **context closure**[2] by setting: $\mathbb{M}\ \tilde{r}\ \mathbb{N}$ iff $\exists\mathbb{C}(\cdot)$ and $M'\ r\ \mathbb{N}'$ s.t. $\mathbb{M} = \mathbb{C}(M'), \mathbb{N} = \mathbb{C}(\mathbb{N}')$.

[2] In [3, 14] bag contexts are defined too, so that context closure extends a relation to $\mathcal{N}\langle\Lambda^{(b)}\rangle \times \mathcal{N}\langle\Lambda^{(b)}\rangle$. In fact we prefer to introduce the term contexts only, making clear that the set $\mathcal{N}\langle\Lambda^r\rangle$ is the actual protagonist of the calculus. However our choice is a matter of taste, affecting no main property of the calculus.

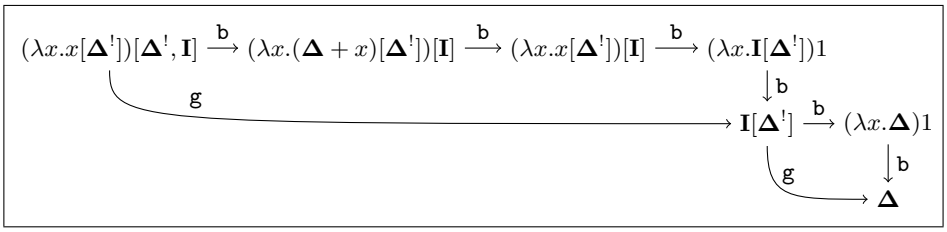

Fig. 3: An example of baby- and giant-step reductions. We use the notation of Fig. 1(b): after the first $\mathtt{b}$-step the term $(\lambda x.(\mathbf{\Delta} + x)[\mathbf{\Delta}^!])[\mathbf{I}]$ stands for $(\lambda x.\mathbf{\Delta}[\mathbf{\Delta}^!])[\mathbf{I}] + (\lambda x.x[\mathbf{\Delta}^!])[\mathbf{I}]$, 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 $(\lambda x.\mathbf{\Delta}[\mathbf{\Delta}^!])[\mathbf{I}] \xrightarrow{\mathtt{b}} 0$.

A context is **linear** if its hole is not under the scope of a $(\,)^!$ operator. Linear contexts can be defined inductively omitting the $M[(C(\cdot))^!]\cdot P$ generation rule in the simple context definition. A **head context** is a context having the hole not in a bag. Head contexts can be defined inductively omitting the rules $M[C(\cdot)]\cdot P$ and $M[(C(\cdot)^!)]\cdot P$ in the simple context definition. Notice that the composition $\mathbb{C}(\mathbb{D}(\cdot))$ of two head (resp. linear) contexts $\mathbb{C}(\cdot)$, $\mathbb{D}(\cdot)$ is head (resp. linear).

We introduce 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 ([3, 14]). *The* **baby-step** *reduction $\xrightarrow{\mathtt{b}}$ is defined by the context closure of the following relation (supposing x not free in N):*

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

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

The **giant-step** *reduction $\xrightarrow{\mathtt{g}}$ is defined by the context closure of the following relation, for $n \geq 0$: $(\lambda x.M)[N_1^{(!)}, \ldots, N_n^{(!)}] \xrightarrow{\mathtt{g}} M\langle\langle N_1^{(!)}/x\rangle\rangle \ldots \langle\langle N_n^{(!)}/x\rangle\rangle\{0/x\}$.*

For any reduction $\xrightarrow{\mathtt{x}}$, we denote by $\xrightarrow{\mathtt{x}+}$ and $\xrightarrow{\mathtt{x}*}$ its transitive and reflexive-transitive closure respectively.

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

Proposition 4 ([3, 14]). *We have $\xrightarrow{\mathtt{g}} \subset \xrightarrow{\mathtt{b}*} \subset \xrightarrow{\mathtt{g}*}\xleftarrow{\mathtt{g}*}$, where the last denotes the 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 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 addend of $(\lambda x.(\mathbf{\Delta} + x)[\mathbf{\Delta}^!])[\mathbf{I}]$).

3 Solvability

Using the λ-calculus terminology, we will call *solvable* the terms representing meaningful programs, i.e., the ones that can interact with the environment. Let us recall that in λ-calculus a term is solvable whenever there is a head context reducing it to the identity [8]. In resource calculus terms appear in formal sums, where repetitions do matter, hence various notions of solvability can arise, depending on the number of times one gets the identity. This paper deals extensively with the weakest notion of solvability, which asks that a term is solvable whenever a suitable context filled with it reduces to a sum, where at least one addend is the identity. This notion is related in some sense to a *may*-semantics of Λ^r, which arises naturally because of the definition of 0 as the neutral element of the sum. However different notions of solvability could be proposed, and we will discuss them in Section 6.

Definition 5. *A term M is* **solvable** *whenever there are a head simple context $C(\cdot)$ and a sum of terms $\mathbb{N}$, possibly 0, such that $C(M) \xrightarrow{\mathtt{g}^*} \mathbf{I} + \mathbb{N}$.*

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

One major outcome of this paper is the characterization of solvability by means of the following notion of head-normalizability (see Theorem 19).

Definition 6. *A term is a* **head-normal form**, hnf *for short, if it has no redex but under the scope of a $(\,)^!$. The set of* hnf *can be defined inductively as follows.*

$$\lambda x.M \text{ is hnf if } M \text{ is hnf},$$
$$xP_1 \dots P_p \text{ is hnf if } p \geq 0 \text{ and } \forall i \leq p, \text{ every linear resource in } P_i \text{ is a hnf}.$$

A sum $\mathbb{M}$ of terms is a head-normal form whenever it contains an addend in head-normal form. A term M is **head-normalizable** *iff it is reducible to a* hnf*.*

Notice that for the resource terms corresponding to λ-terms these notions coincide with the usual ones.

The term $\lambda x.y1[x^!, \mathbf{\Omega}^!]$ is a hnf, and it is solvable via $(\lambda y.(\cdot))[\mathbf{F}]$: indeed, $(\lambda y.(\lambda x.y1[x^!, \mathbf{\Omega}^!]))[\mathbf{F}] \xrightarrow{\mathtt{g}} \lambda x.\mathbf{F}1[x^!, \mathbf{\Omega}^!] \xrightarrow{\mathtt{g}} \lambda x.\mathbf{I}[x^!, \mathbf{\Omega}^!] \xrightarrow{\mathtt{g}} \mathbf{I} + \lambda x.\mathbf{\Omega}$. The terms

$$\frac{}{x:\sigma \vdash x:\sigma}\,{}^{\mathtt{v}} \qquad \frac{}{\vdash 1:\omega}\,{}^{\mathtt{1}} \qquad \frac{\Gamma \vdash A:\pi, \quad \mathbb{B}\neq 0}{\Gamma \vdash A+\mathbb{B}:\pi}\,{}^{\oplus}$$

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

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

Fig. 4: The type assignment system $\vdash$. 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.

$\mathbf{F}[x^!,\mathbf{\Omega}^!]$, $\mathbf{I}[x^!,\mathbf{\Omega}^!]$ are not hnf but both are head-normalizable (the former reducing to $\mathbf{I}$, the latter to $x+\mathbf{\Omega}$); both are clearly solvable, also. The terms $\mathbf{F}[x]$ or $\mathbf{I}[\mathbf{\Omega}^!]$ are not head-normalizable: they reduce to 0 and $\mathbf{\Omega}$, respectively. The notion of head-reduction is extended in this non-deterministic setting as follows.

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

4 An Intersection Type Assignment System

In this section we present an intersection type system assigning types to all and only the expressions having head-normal form (Theorem 19). This system 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 [15], which supplies a logical semantics of the language in [1]. The main logical difference between the two systems is that the one in [15] is affine and describes a lazy operational semantics. In the restricted setting of λ-calculus similar non-idempotent systems have been considered starting from [16], e.g. [17, 18, 19, 12].

Definition 8. *The set of types is the union of the set of* linear types *and that of* intersection types*, given by the following grammars*

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

where the variable a varies on an infinite set of atoms and ω is a constant. We consider types modulo the equivalence $\sim$ generated by the following rules:

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

The last two rules allow us to consider n-ary intersections $\sigma_1\wedge\ldots\wedge\sigma_n$, for any $n\in\mathcal{N}$, 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* $\Gamma \vdash \mathbb{A} : \pi$.

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

Rule $\oplus$ assigns to a sum the type of one of its addends, and it reflects the may-semantics we chose. The condition $\mathbb{B} \neq 0$ is not necessary for characterizing head-normalizable terms, but it is useful to avoid redundant applications of $\oplus$. In the rule $!_n$ the parameter n takes into account the number of times the reusable resource $M^!$ will be called, whereas the rule ℓ assigns just one type to the linear resource M. Note that any bag containing only reusable resources can be typed by ω using the rules $\mathtt{1}$ and $!_0$. All other rules are almost standard.

Let us recall that types are modulo the equivalence $\sim$, which means that all the rules of Figure 4 must be considered closed under the $\sim$.

Definition 9. *The* measure *of a derivation* Φ *is the number* $\mathrm{m}(\Phi)$ *of axioms (i.e.* $\mathtt{v}$ *and* $\mathtt{1}$ *rules) in* Φ. *The measure* $\mathrm{m}(\mathbb{A})$ *of a sum of expressions* $\mathbb{A}$ *is*

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

The following lemmata (Lemma 10-14) state basically that the typing system behaves well with respect to the substitutions of resource calculus. They are needed to prove Proposition 15: typing judgements are invariant under baby and giant-step reductions.

Lemma 10 (Linear Substitution). *Let* $\Phi :: \Gamma, x : \tau \vdash \mathbb{A} : \pi$ *and* $\Psi :: \Delta \vdash N : \tau$. *There is a derivation* $\mathcal{L}(\Phi, \Psi) :: \Gamma, \Delta \vdash \mathbb{A}\langle N/x\rangle : \pi$ *with* $\mathrm{m}(\mathcal{L}(\Phi, \Psi)) = \mathrm{m}(\Phi) + \mathrm{m}(\Psi) - 1$.

Proof (Sketch). By induction on Φ, splitting depending on its last rule. We treat in detail only the case of a terminal $!_n$ rule. The base of induction is trivial: $\mathtt{v}$ is immediate, while $\mathtt{1}$ 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 $\to\mathtt{E}$, ℓ are easier variant of the $!_n$ case. So let us assume

$$\Phi := \dfrac{\begin{array}{ccc}\vdots\ \Phi_i & & \vdots\ \Phi_P\\ \Gamma_i \vdash M : \sigma_i, & \text{for } 1 \le i \le n & \Gamma_P \vdash P : \zeta\end{array}}{\Gamma_1, \ldots, \Gamma_n, \Gamma_P \vdash [M^!]\cdot P : \sigma_1 \wedge \ldots \wedge \sigma_n \wedge \zeta}\,!_n$$

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 an easy variant). Notice that supposing $x : \tau$ in Γ_1 entails $n \ge 1$. By induction there is $\mathcal{L}(\Phi_1, \Psi) :: \Gamma_1', \Delta \vdash M\langle N/x\rangle : \sigma_1$ s.t. $\mathrm{m}(\mathcal{L}(\Phi_1, \Psi)) = \mathrm{m}(\Phi_1) + \mathrm{m}(\Psi) - 1$. Let $M\langle N/x\rangle = \sum_{j=1}^{k} L_j$. In order to be typable $M\langle N/x\rangle$ must have an addend (i.e. $k > 0$), say L_1, and a proof $\Phi_1' :: \Gamma_1', \Delta \vdash L_1 : \sigma_1$ s.t. $\mathrm{m}(\Phi_1') = \mathrm{m}(\mathcal{L}(\Phi_1, \Psi))$. We define

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

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, $\mathrm{m}(\mathcal{L}(\Phi,\Psi)) = \mathrm{m}(\Phi'_1) + \mathrm{m}(\Phi_P) + \sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\mathcal{L}(\Phi_1,\Psi))+\mathrm{m}(\Phi_P)+\sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\Phi_1)+\mathrm{m}(\Psi)-1+\mathrm{m}(\Phi_P)+\sum_{i=2}^{n}\mathrm{m}(\Phi_i) = \mathrm{m}(\Phi)+\mathrm{m}(\Psi)-1$. □

Lemma 11 (Linear Expansion). *Let $\Phi :: \Gamma \vdash \mathbb{A}\langle N/x\rangle : \pi$. There are a linear type τ and derivations $\Phi_1 :: \Gamma_1, x : \tau \vdash \mathbb{A} : \pi$ and $\Phi_2 :: \Gamma_2 \vdash N : \tau$ with $\Gamma = \Gamma_1, \Gamma_2$.*

Proof (Sketch). By structural induction on $\mathbb{A}$, splitting depending on the top level constructor. We detail the case $\mathbb{A} = [M^!]\cdot P$, the other cases being easy variants. If $\mathbb{A}\langle N/x\rangle = [M\langle N/x\rangle, M^!]\cdot P + [M^!]\cdot P\langle N/x\rangle$, then Φ types only one addend of the sum $\mathbb{A}\langle N/x\rangle$ through a $\oplus$ rule. Let us suppose this addend 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 inspecting the rules in Figure 4 one can deduce from Φ a derivation $\overline{\Phi}^1 :: \Gamma^1 \vdash M' : \sigma$ and a derivation $\overline{\Phi}^2 :: \Gamma^2 \vdash [M^!]\cdot P : \overline{\pi}$ 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 $\Gamma^1 \vdash M\langle N/x\rangle : \sigma$, hence by induction hypothesis we have $\overline{\Phi}^1_1 :: \Gamma^1_1, x : \tau \vdash M : \sigma$ and $\overline{\Phi}^1_2 :: \Gamma^1_2 \vdash N : \tau$. Then we define $\Phi_1 :: \Gamma_1, x : \tau \vdash [M^!]\cdot P : \pi$ 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 $\overline{\Phi}^1_2$. □

Lemma 12 (Partial Substitution). *Let $m \geq 0$, $\Phi :: \Gamma, x : \sigma_1, \ldots, x : \sigma_m \vdash \mathbb{A} : \pi$ and $\forall i \leq m$, $\Psi_i :: \Delta_i \vdash N : \sigma_i$ with $\Delta = \Delta_1, \ldots, \Delta_m$. There is $\mathcal{P}(\Phi, \Psi_{i\leq m}) :: \Gamma, \Delta \vdash \mathbb{A}\{(N + x)/x\} : \pi$ with $\mathrm{m}(\mathcal{P}(\Phi, \Psi_{i\leq m})) = \mathrm{m}(\Phi) - m + \sum_{i=1}^{m}\mathrm{m}(\Psi_i)$.*

Proof (Sketch). Like in the proof of Linear Substitution (Lemma 10) we do induction on Φ, splitting depending on its last rule. We detail only the case of a terminal $!_n$ rule, the other cases being immediate or easier variants. So let

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

where $\Gamma = \Gamma_1, \ldots, \Gamma_{n+1}$ and $x : \sigma_1, \ldots, x : \sigma_m = \Gamma_1^x, \ldots, \Gamma_{n+1}^x$. 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_j^x, 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 $\mathcal{P}(\Phi_j, \Psi_{i\in I^j}) :: \Gamma_j, \Delta_{I^j} \vdash M\{N + x/x\} : \tau_j$ for every $j \leq n$, and $\mathcal{P}(\Phi_{n+1}, \Psi_{I^{n+1}}) :: \Gamma_{n+1}, \Delta_{I^{n+1}} \vdash P\{N + x/x\} : \zeta$, 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 every $j \leq n + 1$.

As always, $M\{N + x/x\}$ (resp. $P\{N + x/x\}$) is in general 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. By inspecting the rules of Figure 4, we obtain a function $f : \{0, \ldots, n-1\} \to \{0, \ldots, k-1\}$, an addend P' in $\mathbb{P}$,

and a family of derivations $\Phi'_j :: \Gamma_j, \Delta_{I^j} \vdash M_{f(j)} : \tau_j$ for $j \leq n$, and $\Phi'_{n+1} :: \Gamma_{n+1}, \Delta_{I^{n+1}} \Delta_i \vdash P' : \zeta$, 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 $\mathcal{P}(\Phi, \Psi_{i \leq m}) :=$

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

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

The next lemmata have proofs similar to the previous ones (by induction on $\mathbb{A}$ or Φ). We omit to sketch their proofs.

Lemma 13 (Partial Expansion). *Let $\Phi :: \Gamma \vdash \mathbb{A}\{N + x/x\} : \pi$, then there is a number $m \geq 0$, linear types $\tau_1, \ldots, \tau_m$ and derivations $\Phi_1 :: \Gamma_1, x : \tau_1, \ldots, x : \tau_m \vdash \mathbb{A} : \pi$ and $\Psi_i :: \Delta_i \vdash N : \tau_i$ for $i \leq m$ and $\Gamma = \Gamma_1, \Delta_1, \ldots, \Delta_m$.*

Lemma 14. *Let $x \notin d(\Gamma)$, then for every $\Phi :: \Gamma \vdash \mathbb{A} : \pi$ there is $\Psi :: \Gamma \vdash \mathbb{A}\{0/x\} : \pi$ with $\mathrm{m}(\Phi) = \mathrm{m}(\Psi)$, and vice versa.*

Proposition 15. *Let $\epsilon \in \{\mathtt{b}, \mathtt{g}\}$ and $M \xrightarrow{\epsilon} \mathbb{M}$, then M and $\mathbb{M}$ share the same judgements, i.e. $\Gamma \vdash M : \tau$ iff $\Gamma \vdash \mathbb{M} : \tau$. Also, $M \xrightarrow{\mathtt{og}} \mathbb{M}$ entails $\mathrm{m}(M) > \mathrm{m}(\mathbb{M})$.*

Proof (Scketch). The proof is by structural induction on M. The induction step splits depending on the top-level constructor of M. All cases are easy consequences of the induction hypothesis, 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 one will follow since it corresponds 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. Then m strictly decreases on giant-steps since they correspond to sequences of baby-steps ending always in an empty bag baby-step.

The baby-step has three cases (recall Definition 3), depending on the resource involved in the reduction. The case of the empty bag is proven using Lemma 14,

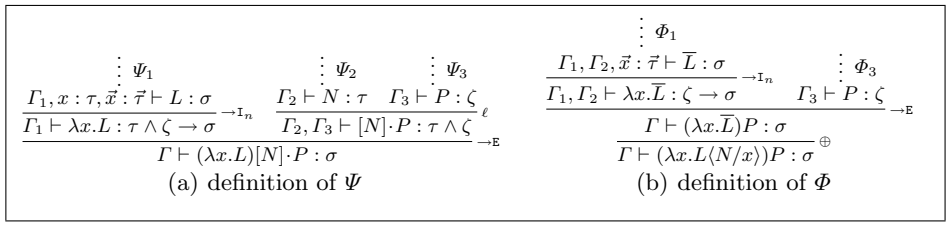

Fig. 5: Definition of the derivations Ψ and Φ used in the proof of Proposition 15.

the case of the bag having one underlined linear resource uses Lemma 10 and 11, and the last case of a bag having one underlined reusable resource uses Lemma 12 and 13. We detail only the linear resource case. Let $M = (\lambda x.L)[N]\cdot P \xrightarrow{\mathtt{b}} (\lambda x.L\langle N/x\rangle)P = \mathbb{M}$ and suppose $\Psi :: \Gamma \vdash (\lambda x.L)[N]\cdot P : \sigma$. By inspecting the rules of Figure 4 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 10) we get $\mathcal{L}(\Psi_1,\Psi_2) :: \Gamma_1, \vec{x}:\vec{\tau}, \Gamma_2 \vdash L\langle N/x\rangle : \sigma$, 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 $\Phi_1 :: \Gamma_1, \vec{x}:\vec{\tau}, \Gamma_2 \vdash \overline{L} : \sigma$ 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 $\Phi :: \Gamma \vdash (\lambda x.L\langle N/x\rangle)P : \sigma$. 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 $\overline{\Phi}_1 :: \Gamma_1, \Gamma_2, \vec{x}:\vec{\tau} \vdash L\langle N/x\rangle : \sigma$. Applying Linear Expansion (Lemma 11) we have $\Psi_1 :: \Gamma_1, \vec{x}:\vec{\tau}, x:\tau \vdash L : \sigma$ and $\Psi_2 :: \Gamma_2 \vdash N : \tau$ (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 $(\lambda x.L)[N]\cdot P$. □

5 Main Theorem

We prove the equivalence among solvability, typability and head-normalizability (Theorem 19). As a byproduct we achieve also an operational characterization through the notion of outer reduction (Definition 7). In various calculi the implication *typable* ⇒ *head-normalizable* is often proven using suitable notions of computability (e.g. [20]) or reducibility candidates (e.g. [21]), whereas the implication *solvable* ⇒ *head-normalizable* is argued through a standardization theorem (e.g. [8]). Our proof is instead based on a different method, namely both implications are easy consequences of Lemma 16, which is argued by induction on the measure on the type derivations given in Definition 9. In the λ-calculus setting, a similar approach can be found in [22]. More in general, the idea of measuring quantitative properties of terms using non-idempotent intersection types can be found also in [12, 23].

Lemma 16. *Let M be a resource term and $C(\cdot)$ be a simple head context. If $C(M)$ is typable, then M is reducible to a* hnf *by outer reduction.*

Proof. We do induction on $\mathrm{m}\big(C(M)\big)$, which is a finite number, being $C(M)$ typable. If M is a hnf we are done. Otherwise it has an outer redex, so let $M \xrightarrow{\mathtt{og}} \mathbb{M}$. Since $C(\cdot)$ is a head context, every outer redex of M is outer in $C(M)$, hence we have $C(M) \xrightarrow{\mathtt{og}} C(\mathbb{M})$. By Proposition 15 $\mathrm{m}\big(C(M)\big) > \mathrm{m}\big(C(\mathbb{M})\big)$. Let $\mathbb{M} = M' + \mathbb{M}''$ be such that $\mathrm{m}\big(C(M')\big) = \mathrm{m}\big(C(\mathbb{M})\big)$: the fact that M' exists is due to $C(\cdot)$ being simple, as every addend in $C(\mathbb{M})$ is obtained by plugging an addend of $\mathbb{M}$ in $C(\cdot)$. By induction hypothesis M' is outer reducible to a hnf $\mathbb{L}$. We conclude by context closure: $M \xrightarrow{\mathtt{og}} M' + \mathbb{M}'' \xrightarrow{\mathtt{og}^*} \mathbb{L} + \mathbb{M}''$. □

Lemma 17. *Every term in head-normal form is solvable.*

Proof. By structural induction on a hnf M. The case $M = \lambda x.M'$ with M' hnf is a trivial consequence of the induction hypothesis. The case $M = xP_1 \dots P_p$ splits in two subcases, depending whether P_1 contains linear resources.
Case I: $P_1 = [L]{\cdot}\overline{P}_1$. We do induction on L and $x\overline{P}_1P_2 \dots P_p$, which are hnf by definition. Thus we obtain two simple head contexts $C(\cdot)$ and $D(\cdot)$ s.t. $C(L) \xrightarrow{\mathtt{g}^*} \mathbf{I}+\mathbb{G}_C$ and $D(x\overline{P}_1P_2 \dots P_p) \xrightarrow{\mathtt{g}^*} \mathbf{I}+\mathbb{G}_D$. Let H be the simple term $\lambda y.C(y)[x[y^!]]$, we have $H[L]{\cdot}\overline{P}_1 \xrightarrow{\mathtt{g}} C(L)[x\overline{P}_1]+\mathbb{G}_H \xrightarrow{\mathtt{g}^*} (\mathbf{I}+\mathbb{G}_C)[x\overline{P}_1]+\mathbb{G}_H \xrightarrow{\mathtt{g}} x\overline{P}_1+\mathbb{G}_C[x\overline{P}_1]+\mathbb{G}_H$, where $\mathbb{G}_H$ is the garbage, possibly 0, generated by putting an element of the bag $\overline{P}_1$ into $C(\cdot)$, instead of L. Then define $E(\cdot) := (\lambda x.(\cdot))[H, x^!]$ and notice $E(x[L]{\cdot}\overline{P}_1P_2 \dots P_p) \xrightarrow{\mathtt{g}} H[L]{\cdot}\overline{P}_1P_2 \dots P_p+\mathbb{G}_E$, where $\mathbb{G}_E$ is the garbage, possibly 0, obtained by linearly substituting H for some free occurrence of x in one P_i's. Finally, we define the context $F(\cdot) := D(E(\cdot))$, which is simple and head, being the composition of simple and head contexts. We have $F(x[L]{\cdot}\overline{P}_1P_2 \dots P_p) \xrightarrow{\mathtt{g}^*} D(H[L]{\cdot}\overline{P}_1P_2 \dots P_p) + \mathbb{G}' \xrightarrow{\mathtt{g}^*} D(x\overline{P}_1P_2 \dots P_p) + \mathbb{G}'' \xrightarrow{\mathtt{g}^*} \mathbf{I} + \mathbb{G}'''$, where, noted in passing, $\mathbb{G}' = D(G_E)$, $\mathbb{G}'' = \mathbb{G}' + D((\mathbb{G}_C[x\overline{P}_1] + \mathbb{G}_H)P_2 \dots P_p)$ and finally $\mathbb{G}''' = \mathbb{G}'' + \mathbb{G}_D$.
Case II: No linear resource in P_1. We do induction on $xP_2 \dots P_p$, thus getting a simple head context $D(\cdot)$ reducing $xP_2 \dots P_p$ to a sum containing the identity. We set $F(\cdot) := (\lambda x.(\cdot))[\lambda y_1 \dots y_p.D(x[y_2^!] \dots [y_p^!]), x^!]$. Easily one checks $F(xP_1P_2 \dots P_p) \xrightarrow{\mathtt{g}^*} D(xP_2 \dots P_p) + \mathbb{G} \xrightarrow{\mathtt{g}^*} \mathbf{I} + \mathbb{G}'$, for suitable $\mathbb{G}$, $\mathbb{G}'$. □

In the λ-calculus the above lemma is trivial since contexts can reduce a head-normal form into the identity simply replacing the head variable with a term erasing all its resources. In the resource calculus this is not possible, because of the linear resources, that cannot be erased but must be used.

Lemma 18. *Every head-normal form is typable.*

Proof. By structural induction on a hnf $\mathbb{M}$. 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} hnf. By induction hypothesis we have derivations $\Psi_{i,j} :: \Gamma_{i,j} \vdash M_{i,j} : \tau_{i,j}$ for each $i \leq p$, $j \leq m_i$ hence we can

construct a derivation $\Phi_i :: \Gamma_{i,1}, \ldots, \Gamma_{i,m_i} \vdash P_i : \tau_{i,1} \wedge \ldots \wedge \tau_{i,m_i}$ by applying a tree of m_i rules ℓ having as premises the $\Psi_{i,1}, \ldots, \Psi_{i,m_i}$ respectively and, as the rightmost leaf, a derivation of $\vdash [N^!_{i,1}, \ldots, N^!_{i,n_i}] : \omega$ made of n_i rules $!_0$ and one rule $\mathtt{1}$. Similarly we get a derivation typing $xP_1 \ldots P_p$ by applying a tree of p rules $\rightarrow_{\mathtt{E}}$ having as premises the Φ_i's derivations and, as the leftmost leaf, a $\mathtt{v}$ rule typing x with $(\bigwedge_{j\leq m_1} \tau_{1,j}) \wedge \ldots \wedge (\bigwedge_{j\leq m_p} \tau_{p,j}) \rightarrow \sigma$, for a linear type σ. □

Theorem 19. *Given a resource term M, the following are equivalent:*

1. *M is head-normalizable,*
2. *M is typable by $\vdash$,*
3. *M is reducible to a* hnf *by outer reduction,*
4. *M is solvable.*

Proof. $1 \Rightarrow 2$: by Prop. 15 and Lemma 18. $2 \Rightarrow 3$: by Lemma 16, merely taking the hole as the simple head context. $3 \Rightarrow 4$: by Lemma 17 and context closure. $4 \Rightarrow 1$: if there is a head simple context $C(\cdot)$ s.t. $C(M)$ has a hnf, by the already proven implication $1 \Rightarrow 2$, $C(M)$ is typable, we conclude by Lemma 16. □

The implication $1 \Rightarrow 3$ can also be argued as a corollary of the standardization proven in [7]. However our proof uses the type assignment system, namely Lemma 16, so it adopts a different approach with respect to the techniques in [7].

6 Concluding Remarks

Theorem 19 achieves a weak operational characterization of solvability. In fact, the non-deterministic nature of the calculus makes terms having different outer reduction sequences, some terminating in a (head-)normal form, others infinite. Take for example $\mathbf{I}[(\mathbf{I}[x])^!, (\mathbf{\Omega}[\mathbf{I}1])^!]$: a first outer step gives $\mathbf{I}[x] + \mathbf{\Omega}[\mathbf{I}1]$, from which starts an outer reduction terminating in x (i.e. $\mathbf{I}[x]+\mathbf{\Omega}[\mathbf{I}1] \xrightarrow{\mathtt{g}} \mathbf{I}[x]+\mathbf{\Omega}[0] = \mathbf{I}[x] \xrightarrow{\mathtt{g}} x$), as well as infinite outer reductions looping on the head-normal form $x + \mathbf{\Omega}[\mathbf{I}1]$ or looping on $\mathbf{I}[x] + \mathbf{\Omega}[\mathbf{I}1]$.

As already mentioned, other notions of solvability are meaningful, depending on the number of times one requires the identity in the resulting sum. The following two seem quite interesting:

- a term M is **must-solvable** whenever there are an applicative simple context $C(\cdot)$ and $n > 0$ such that $C(M) \xrightarrow{\mathtt{g}^*} n\mathbf{I}$;
- a term M is **exactly-solvable** whenever there is an applicative simple context $C(\cdot)$ such that $C(M) \xrightarrow{\mathtt{g}^*} \mathbf{I}$

Clearly exact-solvability implies must-solvability, which in its turn implies the, let us say, *may-solvability* of Definition 5. Also, these three notions do not collapse one another: for example, the term $\mathbf{I}[\mathbf{I}^!, \mathbf{\Omega}^!]$ is may-solvable but not must- nor exactly-solvable, in fact $\mathbf{I}[\mathbf{I}^!, \mathbf{\Omega}^!] \xrightarrow{\mathtt{g}} \mathbf{I}+\mathbf{\Omega}$; the term $\mathbf{I}[\mathbf{I}^!, \mathbf{I}^!]$ is may- and must-solvable, but not exactly-solvable, in fact $\mathbf{I}[\mathbf{I}^!, \mathbf{I}^!] \xrightarrow{\mathtt{g}} 2\mathbf{I}$; the term $\mathbf{I}[\mathbf{I}, \mathbf{I}^!]$ is exactly solvable, hence also may and must solvable, in fact $\mathbf{I}[\mathbf{I}, \mathbf{I}^!] \xrightarrow{\mathtt{g}} \mathbf{I}$. We will give an analysis of all these kinds of solvability in a future work.

Acknowledgements. We are grateful to Rocco De Nicola, Giulio Manzonetto, Mauro Piccolo and Paolo Tranquilli for useful discussions and hints.

References

[1] Boudol, G.: The Lambda-Calculus with Multiplicities. INRIA Report 2025 (1993)
[2] Ehrhard, T., Regnier, L.: The Differential Lambda-Calculus. Theor. Comput. Sci. **309**(1) (2003) 1–41
[3] Tranquilli, P.: Intuitionistic Differential Nets and Lambda-Calculus. Theor. Comput. Sci., to appear (2008)
[4] de'Liguoro, U., Piperno, A.: Non Deterministic Extensions of Untyped Lambda-Calculus. Inf. Comput. **122**(2) (1995) 149–177
[5] Ehrhard, T., Regnier, L.: Böhm trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms. In: CiE. Volume 3988 of LNCS. (2006) 186–197
[6] Ehrhard, T., Regnier, L.: Uniformity and the Taylor Expansion of Ordinary Lambda-Terms. Theor. Comput. Sci. **403**(2-3) (2008) 347–372
[7] Pagani, M., Tranquilli, P.: Parallel Reduction in Resource Lambda-Calculus. In: APLAS. Volume 5904 of LNCS. (2009) 226–242
[8] Barendregt, H.: The Lambda-Calculus, its Syntax and Semantics. Second edn. Number 103 in Stud. Logic Found. Math. North-Holland (1984)
[9] Coppo, M., Dezani-Ciancaglini, M., Venneri, B.: Functional Characters of Solvable Terms. Zeitschrift für Mathematische Logik **27** (1981) 45–58
[10] Hyland, J.M.E.: A Syntactic Characterization of the Equality in Some Models of the Lambda Calculus. J. London Math. Soc. **2**(12) (1976) 361–370
[11] Ronchi Della Rocca, S., Paolini, L.: The Parametric λ-Calculus: a Metamodel for Computation. EATCS Series. Springer, Berlin (2004)
[12] de Carvalho, D.: Execution Time of λ-Terms via Denotational Semantics and Intersection Types. Submitted for publication (2009)
[13] Bucciarelli, A., Ehrhard, T., Manzonetto, G.: Not Enough Points Is Enough. In: CSL. Volume 4646 of Lecture Notes in Comp. Sci. (2007) 298–312
[14] Tranquilli, P.: Nets between Determinism and Nondeterminism. Ph.D. thesis, Università Roma Tre/Université Paris Diderot (Paris 7) (April 2009)
[15] Boudol, G., Curien, P.L., Lavatelli, C.: A Semantics for Lambda Calculi with Resources. MSCS **9**(5) (1999) 437–482
[16] Coppo, M., Dezani-Ciancaglini, M., Venneri, B.: Principal Type Schemes and Lambda-Calculus Semantics. In: To H. B. Curry. Essays on Combinatory Logic, Lambda-calculus and Formalism, Accademic Press (1980) 480–490
[17] Kfoury, A.J.: A Linearization of the Lambda-Calculus and Consequences. Journal of Logic and Computation **10**(3) (2000) 411–436
[18] Wells, J.B., Dimock, A., Muller, R., Turbak, F.: A Calculus with Polymorphic and Polyvariant Flow Types. J. Funct. Program. **12**(3) (2002) 183–227
[19] Neergaard, P.M., Mairson, H.G.: Types, Potency, and Idempotency: why Nonlinearity and Amnesia Make a Type System Work. In: ICFP, ACM (2004) 138–149
[20] Coppo, M., Dezani-Ciancaglini, M., Zacchi, M.: Type Theories, Normal Forms and D_∞-Lambda-Models. Inf. Comput. **72**(2) (1987) 85–116
[21] Girard, J.Y.: Interprétation Fonctionnelle et Élimination des Coupures de l'Arithmétique d'Ordre Supérieur. Thèse de doctorat, Université Paris 7 (1972)
[22] Valentini, S.: An elementary proof of strong normalization for intersection types. Archive for Mathematical Logic **40**(7) (October 2001) 475–488
[23] de Carvalho, D., Pagani, M., Tortora de Falco, L.: A Semantic Measure of the Execution Time in Linear Logic. Theor. Comput. Sci., to appear (2008)