Strong Normalization from an unusual point of view ⋆

Luca Paolini

Università di Torino (ITALY), Dipartimento di Informatica

Elaine Pimentel

Universidade Federal de Minas Gerais (BRAZIL), Departamento de Matemática

Simona Ronchi Della Rocca

Università di Torino (ITALY), Dipartimento di Informatica

Abstract

A new complete characterization of β-strong normalization is given, both in the classical and in the lazy λ-calculus, through the notion of potential valuability inside two suitable parametric calculi.

Key words: strong normalization, call-by-value, parametric lambda calculus

1 Introduction

The notion of β-normal form and consequently of β-strong normalization is central in λ-calculus. In fact, Böhm's Theorem [4] says that two extensionally different β-normal forms cannot be equated in any model. As far as we know, there are three characterizations of β-strong normalization: operational, logical and semantical. The operational characterization is due to Barendregt, through the definition of a perpetual reduction strategy. Namely, a term is strongly normalizing if and only if a perpetual reduction strategy, when applied to it, eventually stops, and he gave an example of an effective perpetual strategy [3, (pag. 338)]. The logical characterization has been given through intersection types: the core intersection types system assign types to all and

⋆ Paper partially supported by FAPEMIG and CNPq.

only the strongly normalizing λ-terms. The system is due to Coppo and Dezani [6], while the strong normalization proof has been done by Pottinger [17]. The semantical characterization is due to Coppo, Dezani and Zacchi [5], who designed a filter λ-model, such that the interpretation of λ-term in it is greater than a given point if and only if the term itself is β-strongly normalizing.

The λ-calculus models a call-by-name evaluation. If we skitch to call-by-value and consider the $\lambda\beta_v$-calculus defined by Plotkin [16] in order to model it, the notion of normal form (and so the notion of strong normalization) becomes meaningless. In fact, it has been proved [18] that in the $\lambda\beta_v$-calculus, there are normal forms that can be consistently equated. An important notion in this calculus is that of *potential valuability*: a term is potentially valuable if and only if there is a substitution, replacing variables by values, such that the resulting term reduces to a value. A witness of the semantic meaning of such a class of terms is the fact that all non potentially valuable terms can be consistently equated [18]. Different call-by-value λ-calculi can be defined, by modifying in a suitable way the definition of values and consequently obtaining different characterizations of potential valuability. So it is natural to ask if there is some formal relation between the notions of strong normalization and the one of potential valuability. Namely, we consider the notion of β-strong normalization both in the classical λ-calculus and in the lazy λ-calculus (the λ-calculus with a weak β-reduction[1]). The relevant question then is if the set of β-strongly normalizing terms in such calculi, corresponds to that of potential valuability in some call-by-value version of λ-calculus. A positive answer gives a further characterization of the notion of strong normalization, both in the standard and in the lazy cases. To be more precise, we are interested in the following problems:

Problem 1: Is there a call-by-value λ-calculus, such that the set of its potentially valuable terms coincides with the set of β-strongly normalizing terms?

Problem 2: Is there a call-by-value λ-calculus, such that the set of its potentially valuable terms coincides with the set of the weak β-strongly normalizing terms?

In order to deal with theses two problems, we use as syntactical tool the parametric λ-calculus, defined in [15,18], which allows us to deal with some different calculi sharing the same syntax, but whose reduction rules are restrictions of the β-rule. These reduction rules can be fired only when the arguments belong to a particular set of terms, called input values, satisfying some requirements. Both the classical λ-calculus and the $\lambda\beta_v$-calculus are particular instantiations of the parametric λ-calculus, with sets of input values Λ and Γ respectively (see Section 2). We will denote by $\lambda\Delta$-calculus the calculus whose set of input values is Δ.

In this paper, we prove that both problems listed above have a solution. In fact, it is easy to see that they have trivial solutions (Section 3). But we seek for *the best* solutions, meaning minimal and decidable ones. For the first one, we prove that there is a minimal solution, in the sense that there is a set of input values Φ such that the language defined by it satisfies Problem 1 and, moreover, it is minimal between all sets of input values satisfying the same problem. The set Φ is not recursive, but we prove that there is not a decidable solution to this problem. For Problem 2, we prove that the set of input values of the $\lambda\beta_v$-calculus is a solution. It is decidable but not minimal. We prove that a minimal solution does not exist.

We use as technical tools for proving these results the intersection types and the reducibility method, based on an unusual definition of saturated sets. In particular, we will define an intersection type assignment system, characterizing the β-strong normalization in the lazy λ-calculus, so supplying as independent result a logical charactization of lazy β-strong normalization.

Some partial results of this paper have been already presented in preliminary works. Problem 1 and its solution Φ have been presented in [13], and Problem 2 has been discussed in [12]. Issues of minimality and decidability are discussed here for the first time. Moreover, the definition of saturated set given in the present paper is new, and it provides very compact and uniform proofs for both problems.

The paper is organized as follows. Section 2 contains an introduction to the Parametric λ-calculus, Section 3 states the two problems we want to solve, and proposes a solution for each one. In Section 4 two intersection type assignment systems are presented. Section 5 contains the proofs that, for both the considered calculi, strong normalization implies typability (in a different type assignment system). In Section 6 we complete the proofs, showing that typability implies strong normalization. Section 7 contains a discussion about the minimality and decidability of the given solution and the related proofs.

2 The Parametric λ-Calculus

A calculus is a language equipped with some reduction rules. We will consider here calculi sharing the same language, the language of λ-calculus, while they differ from each other in the use of the reduction rule. In order to treat them in a uniform way we will use the notion of parametric calculus, the $\lambda\Delta$-calculus, that gives rise to different calculi by different instantiations of the parameter Δ. The $\lambda\Delta$-calculus has been studied in [15,18]. We use the terminology of [3,18].

Definition 1 (The language Λ)
Let Var *be a countable set of variables. The set Λ of λ-terms is defined by the following grammar:*

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

λ-terms will be ranged over by Latin capital letters. Sets of λ-terms will be denoted by Greek capital letters. $FV(M)$ denotes the set of variables occurring free in the term M, and a term M is said closed if $FV(M) = \emptyset$. If Θ denotes a set of terms $(\Theta)^0$ is the set of closed terms belonging to Θ.

Sometimes, we will refer to λ-terms simply as terms. As usual, terms will be considered modulo α-conversion, i.e., modulo names of bound variables. The symbol $\equiv$ will denote syntactical identity of terms, up to α-equivalence.

We will use the following abbreviations, in order to avoid an excessive number of parentheses, thereby $\lambda x_1...x_n.M$ will stand for $(\lambda x_1(...(\lambda x_n.M)...))$ and $MN_1N_2...N_n$ will stand for $(...((MN_1)N_2)...N_n)$. Moreover $|M|$ will be used in order to denote the number of symbol of the term M.

The $\lambda\Delta$-calculus consists of the language Λ equipped with a set $\Delta \subseteq \Lambda$ of input values, satisfying some closure conditions. Informally, input values represent already evaluated terms, that can be passed as arguments. The set Δ of input values and the reduction $\rightarrow_\Delta$, induced by it, are defined below.

Definition 2 *Let $\Delta \subseteq \Lambda$.*

(i) *The* Δ-reduction *($\rightarrow_\Delta$) is the contextual closure of the following rule:*

$$(\lambda x.M)N \rightarrow M[N/x] \text{ if and only if } N \in \Delta.$$

$(\lambda x.M)N$ is a Δ-redex *(or simply redex).*

(ii) *$\rightarrow_\Delta^+$, $\rightarrow_\Delta^*$ and $=_\Delta$ are respectively the transitive closure of $\rightarrow_\Delta$, the reflexive and transitive closure of $\rightarrow_\Delta$ and the symmetric, reflexive and transitive closure of $\rightarrow_\Delta$.*

(iii) *A set $\Delta \subseteq \Lambda$ is a* set of input values*, when the following conditions are satisfied:*

- Var $\subseteq \Delta$ *(*Var-closure*);*
- *$P, Q \in \Delta$ implies $P[Q/x] \in \Delta$, for each $x \in$ Var* *(*substitution closure*);*
- *$M \in \Delta$ and $M \rightarrow_\Delta N$ imply $N \in \Delta$* *(*reduction closure*).*

The closure conditions on the set of input values assure us that $\lambda\Delta$-calculus has the confluence property for every Δ , i.e., the following theorem holds.

Theorem 3 (Confluence) *[15,18] Let $M \rightarrow_\Delta^* N_1$ and $M \rightarrow_\Delta^* N_2$.*
There is Q such that both $N_1 \rightarrow_\Delta^* Q$ and $N_2 \rightarrow_\Delta^* Q$.

Two particular instantiations of Δ give rise to the call-by-name and the call-by-value λ-calculus. The call-by-name λ-calculus (i.e., the standard λ-calculus equipped with the β-reduction) coincides with the $\lambda\Lambda$-calculus. The standard call-by-value λ-calculus (defined by Plotkin in [16]) coincides with the $\lambda\Gamma$-calculus, where $\Gamma = \mathrm{Var} \cup \{\lambda x.M \mid M \in \Lambda\}$.

Definition 4 *Let Δ be a set of input values.*

(i) *A term of the $\lambda\Delta$-calculus is in* Δ-normal form *if and only if it does not contain occurrences of Δ-redexes.*
(ii) *A term M is* strongly Δ-normalizing *if every reduction sequence starting from M eventually stops.*

Let Δ-NF and Δ-SN denote respectively the set of Δ-normal forms and of Δ-strongly normalizing terms. The set Δ-NF can be defined in the following recursive way:

$$\begin{aligned}\Delta\text{-NF} = \mathrm{Var} &\cup \{xM_1...M_n \mid M_k \in \Delta\text{-NF } (1 \leq k \leq n)\} \\ &\cup \{\lambda\vec{x}.M \mid M \in \Delta\text{-NF}\} \\ &\cup \{(\lambda x.P)QM_1...M_n \mid P, Q, M_k \in \Delta\text{-NF}, Q \notin \Delta \quad (1 \leq k \leq n)\}.\end{aligned}$$

Note that for the $\lambda\Lambda$-calculus, being Λ its set of input values, the last case cannot happen, i.e., there are no normal forms of the shape $(\lambda x.P)QM_1...M_n$, hence Λ-NF$\subseteq \Delta$-NF, for all Δ. But it can happen for the $\lambda\Gamma$-calculus: indeed $\lambda uv.(\lambda x.x)(uv)$ is a Γ-normal form.

In the $\lambda\Gamma$-calculus, the notion of normal form is meaningless. In fact, there are different Γ-normal forms that can be consistently equated [11,18]. The key notion in a call-by-value setting is the one of (potential) valuability, given by the next definition (see [14,18]).

Definition 5

(i) *A term M is* Δ-valuable *if and only if there is $N \in \Delta$ such that $M \to_\Delta^* N$.*
(ii) *A term M is* potentially Δ-valuable *if and only if there is a substitution $\mathbf{s}$, replacing variables by terms belonging to Δ, such that $\mathbf{s}(M)$ is Δ-valuable.*

Let Δ-PV be the set of Δ-potentially valuable terms.

It is immediate to verify that a closed term is in Δ-PV if and only if it is Δ-valuable. Note that the notion of Δ-normal form and that one of potentially Δ-valuable are orthogonal. As an example, consider the $\lambda\Gamma$-calculus, and the term $M \equiv (\lambda z.D)(yI)D$, where $I \equiv \lambda x.x$ and $D \equiv (\lambda z.zz)$. M is in Γ-normal form, but it is neither an input value nor potentially Γ-valuable. In fact, consider $M[Q/y]$, for some $Q \in (\Gamma)^0$. If QI reduces to an element in Γ

then $M[Q/y] \equiv (\lambda z.D)(QI)D$ reduces to DD, which is not an input value. Otherwise $M[Q/y] \to^*_\Gamma (\lambda z.D)Q'D$, for every Q' such that $QI \to^*_\Gamma Q'$, which is not an input value. Thus $(\lambda z.D)(QI)D$ is not Γ-valuable. We call Δ-*liar-normal forms* terms which are in Δ-normal form but are not potentially Δ-valuable.

In the $\lambda\Lambda$-calculus, the notion of solvability plays an important role since, in some sense, solvable terms represent meaningful computations [3]. In [14] the Γ-solvable and potentially Γ-valuable terms has been characterized. This notion has been extended to the parametric $\lambda\Delta$-calculus in [18].

2.1 Lazy reduction

The evaluation of a λ-term is said *lazy* if no reduction is made under the scope of a λ-abstraction. It is possible to define directly the lazy reduction (sometimes called weak), as shown in the next definition.

Definition 6 *Let Δ be a set of input values. The* $\Delta\ell$-reduction *is the applicative closure of the Δ-rule. We will denote by $\to_{\Delta\ell}$ the $\Delta\ell$-reduction, by $\to^*_{\Delta\ell}$ its reflexive and transitive closure, and by $=_{\Delta\ell}$ its symmetric, reflexive and transitive closure.*

Notice that the definition of $\Delta\ell$-reduction is not standard. In fact, the reduction is defined by closing the reduction rule only under application, while in the standard case the closure is under abstraction too.

The notion of normal form can be adapted for the $\Delta\ell$-reduction in the obvious way, as shown in the next definition. Informally a term is in $\Delta\ell$-normal form if it has no occurrences of Δ-redexes, but under the scope of a λ-abstraction.

Definition 7 (i) *A term M* is in $\Delta\ell$-normal form *if and only if it has no occurrences of $\Delta\ell$-redexes.*
(ii) *A term M* has $\Delta\ell$-normal form *if and only if there is a term N in lazy Δ-normal form such that $M \to^*_{\Delta\ell} N$.*
(iii) *A term M is* $\Delta\ell$-strongly normalizing *if and only if there is not an infinite sequence of $\Delta\ell$-reductions starting from it.*

Clearly, a Δ-normal form is a $\Delta\ell$-normal form.

Note that the $\Lambda\ell$-normal form of a term, if there exists, may not be unique. In fact, $(\lambda xy.x)(II) \to^*_{\Lambda\ell} \lambda y.II$ and $(\lambda xy.x)(II) \to^*_{\Lambda\ell} \lambda y.I$ where both $\lambda y.II$ and $\lambda y.I$ are $\Lambda\ell$-normal forms.

We will now consider two particular calculi, namely the Λ-calculus and the $\Lambda\ell$-calculus. We will prove some useful properties related to their strong normalization.

Lemma 8 (i) *$M \in \Lambda$-SN implies that all subterms of M belong to Λ-SN.*
(ii) *$M \notin \Lambda$-SN implies that $\mathbf{s}(M) \notin \Lambda$-SN, for all substitutions $\mathbf{s}$.*
(iii) *$C[(\lambda x.P)Q] \notin \Lambda$-SN, $Q \in \Lambda$-SN and $C[(\lambda x.P)Q] \to_\Lambda C[P[Q/x]]$ imply $C[P[Q/x]] \notin \Lambda$-SN.*

PROOF.

(i) Obvious.
(ii) It is sufficient to observe that every substitution preserves an infinite reduction chain.
(iii) By definition $C[(\lambda x.P)Q] \notin \Lambda$-SN implies that there is an infinite reduction chain starting from it. Since $Q \in \Lambda$-SN, the infinite reduction sequence is preserved by the reduction. □

Lemma 9 (i) *$M \notin \Lambda\ell$-SN implies that $\mathbf{s}(M) \notin \Lambda\ell$-SN, for every substitution $\mathbf{s}$.*
(ii) *$C[(\lambda x.P)Q] \notin \Lambda\ell$-SN, $Q \in \Lambda\ell$-SN and $C[(\lambda x.P)Q] \to_{\Lambda\ell} C[P[Q/x]]$ imply $C[P[Q/x]] \notin \Lambda\ell$-SN.*

PROOF. Similar to the proof of Lemma 8.(ii) and (iii). □

3 The problems and the proposed solutions

Due to the introduction of the parametric λ-calculus, we can rephrase in a more precise way the two problems stated in the introduction.

Problem 1 Is there a set of input values Δ such that the set of potentially Δ-valuable terms coincides with the set of Λ-strongly normalizing terms?
Problem 2 Is there a set of input values Δ such that the set of potentially Δ-valuable terms coincides with the set of $\Lambda\ell$-strongly normalizing terms?

First of all, one could think of proposing Λ-SN and $\Lambda\ell$-SN as trivial solutions for the two problems, respectively. But both sets are not sets of input values, since they do not have the closure properties of Definition 2: they are closed

under Λ and $\Lambda\ell$-reductions respectively, but they are not closed under substitution. Indeed, they would induce non-confluent calculi [15, Theorem 28]. An easy way to restrict these sets in order to satisfy also this constraint is to take only the closed terms. Remembering that a set of input values needs to contain the set of variables, we can define the two subsets of Λ-SN and $\Lambda\ell$-SN: $\Delta_0 = (\Lambda\text{-SN})^0 \cup \text{Var}$ and $\Delta_1 = (\Lambda\ell\text{-SN})^0 \cup \text{Var}$. It is easy to check that Δ_0, Δ_1 are sets of input values and that they are solutions to Problems 1 and 2, respectively.

Theorem 10 (i) Δ_0-$PV = \Lambda$-SN.
(ii) Δ_1-$PV = \Lambda\ell$-SN.

PROOF.

(i) $M \in \Delta_0$-PV implies, by definition, that there is a substitution $\mathbf{s} : Var \to \Delta_0$ such that $\mathbf{s}(M) \to^*_{\Delta_0} N \in \Delta_0$, which implies that N is Λ-strongly normalizing. All redexes in the considered reduction sequence have arguments in $\Delta_0 \subset \Lambda$-SN, so $\mathbf{s}(M) \in \Lambda$-SN by Lemma 8.(iii). Hence $M \in \Lambda$-SN, by Lemma 8.(ii).
(ii) Similar to that of Lemma 9.(i). □

The previous theorem shows that both the sets Λ-SN and $\Lambda\ell$-SN can be described through a proper subset of them. Hence a more interesting question would be:

Problem 3 Are there two minimal restrictions of Λ-SN and $\Lambda\ell$-SN which are solutions of Problem 1 and 2 respectively?

Moreover, it is well known that both Λ-SN and $\Lambda\ell$-SN are not recursive sets, and hence neither are Δ_0 and Δ_1. Thus another relevant question would be:

Problem 4 Are there two recursive sets of input values which are solutions of Problem 1 and 2 respectively?

Let us first consider the set of Λ-strongly normalizing terms. A first attempt on finding a decidable solution, and hence solving Problem 4, would be to take, as a decidable restriction of Λ-SN, the set Λ-NF, trasforming it into a set of input values by taking the subset of its closed terms plus the variables, i.e., $\Delta_2 = (\Lambda\text{-NF})^0 \cup \text{Var}$. But this does not work, since, for example, $I(\lambda x.(I(xx))$ is Λ-strongly normalizing, it is closed, but it does not Δ_2-reduce to a term in Δ_2. In fact both xx and $I(xx)$ do not belong to Δ_2.

We will prove in Section 7 that there isn't a decidable set of input values, which is a proper subset of Δ_0 *and* a solution to Problem 1. Next definition is

of the set of terms Φ, that will be proved to bet a minimal solution to Problem 1, and hence solving Problem 3 restricted to Λ-SN.

Definition 11 (i) *The sets of λ-terms Υ_i, Φ_i ($i \in \mathbb{N}$) are defined by mutual induction, as follows*

$$\begin{aligned}
\Upsilon_0 &= \mathrm{Var} \\
\Phi_i &= \mathrm{Var} \cup (\Upsilon_i)^0 \\
\Upsilon_{i+1} &= \mathrm{Var} \cup \{xM_1...M_n \mid M_k \in \Upsilon_i (1 \leq k \leq n)\} \cup \{\lambda\vec{x}.M \mid M \in \Upsilon_i\} \\
&\quad \cup \left\{ (\lambda x.P)QM_1...M_n \;\middle|\; \begin{array}{l} Q \in \Upsilon_i - (\Lambda^0 \cup \mathrm{Var}), \quad M_1, ..., M_n \in \Upsilon_i, \\ P[Q/x]M_1 \ldots M_n \rightarrow^*_{\Phi_i} R \in \Upsilon_i \end{array} \right\}
\end{aligned}$$

(ii) $\Upsilon = \cup_i \Upsilon_i$ *and* $\Phi = \mathrm{Var} \cup (\Upsilon)^0$.

For example, $\Phi_0 = \mathrm{Var}$, $\Upsilon_1 = \mathrm{Var} \cup \{xy_1...y_n \mid y_i \in Var\} \cup \{\lambda\vec{x}.y \mid y \in \mathrm{Var}\}$ and $\Phi_1 = \mathrm{Var} \cup \{\lambda x_1...x_m.x_j \mid 1 \leq j \leq m\}$.

Proposition 12 *Φ_i is a set of input values, for all $i \in \mathbb{N}$.*

However, let us notice that neither Υ nor Υ_i are sets of input values.

The motivation behind the contruction of set Φ is on what follows. Φ is the least solution of the two recursive equations

$$\begin{aligned}
\Theta = \mathrm{Var} &\cup \{xM_1...M_n \mid M_k \in \Theta \ (1 \leq k \leq n)\} \\
&\cup \{\lambda\vec{x}.M \mid M \in \Theta\} \cup \\
&\{(\lambda x.P)QM_1...M_n \mid P, Q, M_k \in \Theta \ (1 \leq k \leq n), Q \notin Var \cup (\Lambda)^0, \\
&P[Q/x]M_1...M_n \rightarrow^*_\Sigma N \in \Sigma\} \\
\Sigma = Var &\cup (\Theta)^0
\end{aligned}$$

The previous equations characterize the minimal set Δ satisfying the three following constaints:

- $\Delta \subseteq \Delta$-NF.
- Δ is a set of input values.
- Δ does not contain terms that can be equated to Δ-liar-normal forms.

In fact the set Φ satisfies some other properties, as given in the next proposition.

Lemma 13 (i) $\Phi \subset \Phi$-*NF ;*
(ii) Φ *is a set of input values;*

(iii) $M \in \Phi$ *implies* M *is potentially* Φ*-valuable;*
(iv) $\Phi \subset \Lambda$*-SN.*

PROOF.

(i) By construction, $\Phi \subseteq \Phi$-NF . A counterexample proving that the inclusion is proper is the term $M \equiv \lambda z.(\lambda x.D)(zI)D$. $M \in \Phi$-NF since $zI \notin \Phi$. But $M \notin \Upsilon$ and $M \notin \Phi$.
(ii) By construction Φ contains the variables. Moreover a term in Φ either is a variable or it is closed, so the substitution property is satisfied. The reduction closure follows from the previous point.
(iii) $M \in \Phi$ implies, by construction, that either M is a variable or it is closed. So M is potentially Φ-valuable if and only if it is Φ-valuable, and this last property follows immediately from point [(i)].
(iv) The inclusion is obvious. As far as the proper inclusion, $II \in \Lambda$-SN but it does not belong to Φ. □

Lemma 13.(iii) implies that the $\lambda\Phi$-calculus enjoys the confluence property. Moreover it is possible to check that it also satisfies the additional necessary condition for standardization, stated in [15,18].

As far as Problem 2 is concerned, we will prove, in Section 7, that there is not a minimal solution. On the other hand, a decidable set of input values satifying the Problem 2 is $\Gamma = \mathrm{Var} \cup \{\lambda x.M \mid M \in \Lambda\}$, which is the set of input values of Plotkin's $\lambda\beta_v$-calculus [16].

4 Two intersection type assignment systems

In this section, we introduce two type assignment systems, assigning to λ-terms intersection types, which will be the fundamental tools for proving our results. The first one is the system already introduced in [6], while the second one has been defined in [13].

Definition 14 (i) *Let* C_ν *be a countable set of* type-constants *(ranging over* $\alpha, \beta, ..$*) containing at least the type constant* ν *and let* $\mathsf{C} = \mathsf{C}_\nu - \{\nu\}$.
The set $T(\mathsf{C})$ *of* types*, ranging over by* $\sigma, \tau, \pi, \rho, ..$ *is inductively defined as follows:*

$$\begin{aligned} \sigma \in \mathsf{C} \quad &\Rightarrow \quad \sigma \in T(\mathsf{C}) \\ \sigma, \tau \in T(\mathsf{C}) \quad &\Rightarrow \quad (\sigma \to \tau) \in T(\mathsf{C}) \\ \sigma, \tau \in T(\mathsf{C}) \quad &\Rightarrow \quad (\sigma \wedge \tau) \in T(\mathsf{C}). \end{aligned}$$

$T(\mathsf{C}_\nu)$ is defined similarly. Types will be considered modulo associativity, commutativity and idempotency of the constructor $\wedge$ (i.e., modulo an equivalence $\simeq$ which is the contextual, reflexive and transitive closure of the following rules: $\sigma\wedge\tau \simeq \tau\wedge\sigma$, $\sigma \simeq \sigma\wedge\sigma$ and $(\sigma\wedge\tau)\wedge\pi \simeq \sigma\wedge(\tau\wedge\pi)$). We use the convention we will use the notation $\sigma_1\wedge ...\wedge\sigma_n$ for denoting every type up to $\simeq$. Moreover the constructor $\wedge$ take precedence over $\to$.

(ii) *A basis (or ν-basis) is a partial function from* Var *to $T(\mathsf{C})$ (or from* Var *to $T(\mathsf{C}_\nu)$) having a finite domain of definition. If B is a (ν)-basis then $B[\sigma/x]$ denotes the (ν)-basis such that*

$$B[\sigma/x](y) = \begin{cases} \sigma & \text{if } y \equiv x, \\ B(y) & \text{otherwise.} \end{cases}$$

Furthermore, the (ν)-basis B such that $dom(B) = \{x_1, ..., x_n\}$ and $B(x_i) = \sigma_i$, for $1 \leq i \leq n$ will be denoted by $[\sigma_1/x_1, ..., \sigma_n/x_n]$.

(iii) *The* type assignment system $\vdash$ *is a formal system proving typing judgments of the shape:*

$$B \vdash M : \sigma$$

where M is a term, $\sigma \in T(\mathsf{C})$ and B is a basis.

The type assignment system $\vdash$ consists of the following rules:

$$\frac{}{B[\sigma/x] \vdash x : \sigma}\ (var)$$

$$\frac{B[\sigma/x] \vdash M : \tau}{B \vdash \lambda x.M : \sigma \to \tau}\ (\to I) \qquad \frac{B \vdash M : \sigma \to \tau \quad B \vdash N : \sigma}{B \vdash MN : \tau}\ (\to E)$$

$$\frac{B \vdash M : \sigma \quad B \vdash M : \tau}{B \vdash M : \sigma \wedge \tau}\ (\wedge I)$$

$$\frac{B \vdash M : \sigma \wedge \tau}{B \vdash M : \sigma}\ (\wedge E_l) \qquad \frac{B \vdash M : \sigma \wedge \tau}{B \vdash M : \tau}\ (\wedge E_r)$$

(iv) *The* type assignment system $\vdash_\nu$ *is a formal system proving typing judgments of the shape:*

$$B \vdash_\nu M : \sigma$$

where M is a term, $\sigma \in T(\mathsf{C}_\nu)$ and B is a ν-basis.

The type assignment system $\vdash_\nu$ consists of the same rules for $\vdash$ plus the rule:

$$\frac{}{B \vdash_\nu \lambda x.M : \nu}\ (\nu)$$

We will write $\vdash_*$ when referring to both $\vdash$ and $\vdash_\nu$, and we will use the word basis to refer both to basis and ν-basis.

If B, B' are bases then $B \cap B'$ is the basis defined as follows:

$$(B \cap B')(y) = \begin{cases} B(y) \wedge B'(y) & \text{if both } B(y) \text{ and } B'(y) \text{ are defined,} \\ B(y) & \text{if } B(y) \text{ is defined and } B'(y) \text{ is undefined,} \\ B'(y) & \text{if } B'(y) \text{ is defined and } B(y) \text{ is undefined,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The following lemma relates the shape of a term with the shape of its typing derivation.

Lemma 15 (Generation)

(i) *If* $B \vdash_* M : \sigma$, *then* $B - \{x : \tau \mid x \notin FV(M)\} \vdash_* M : \sigma$.
(ii) *If* $B \vdash_* M : \sigma$ *then* $B \cap B' \vdash_* M : \sigma$, *for any basis* B'.
(iii) *If* $B \vdash_* x : \sigma$ *then either* $x : \sigma \in B$ *or* $x : \rho \in B$, *where* $\rho \simeq \sigma \wedge \tau$, *for some* τ.
(iv) *If* $B \vdash_* MN : \sigma$ *then there are types* ρ_i *and* τ_i *such that* $\sigma \simeq \rho_1 \wedge \ldots \wedge \rho_n$, $B \vdash_* M : \tau_i \to \rho_i$ *and* $B \vdash_* N : \tau_i$ *with* $1 \leq i \leq n$.
(v) $B \vdash_* \lambda x.M : \sigma \to \tau$ *if and only if* $B[\sigma/x] \vdash_* M : \tau$.

PROOF.

(i) Trivial.
(ii) Easy, by induction on the derivation d proving $B \vdash_* M : \sigma$.
(iii) Easy, by induction on the derivation and remembering the definition of $\simeq$.
(iv) Easy, by induction on the derivation d proving $B \vdash_* MN : \sigma$.
(v) ($\Leftarrow$) By rule $(\to I)$.
($\Rightarrow$) $d : B \vdash \lambda x.M : \sigma \to \tau$ implies that d has the following shape. There are $k \geq 1$ subderivations d_i of d, ending by:

$$\frac{B[\sigma_i/x] \vdash M : \tau_i}{B \vdash \lambda x.M : \sigma_i \to \tau_i}\,(\to I)$$

followed by a sequence of applications of rules $(\wedge I)$ and $(\wedge E)$, being these ones the only rules that do not modify the shape of the term. Then $\sigma \equiv \sigma_i$ and $\tau \equiv \tau_i$, for some i, and the proof is given. The case for the system $\vdash_\nu$ is similar, taking into account that some of the d_i can end with an application of the rule (ν), but clearly each occurrence of ν will be erased in the rest of the proof. $\square$

The following, very easy, property puts in evidence the difference between the two type assignment systems, and will be useful in what follows.

Proposition 16 (i) *$d : B \vdash M : \sigma$ implies that every subterm of M is typed by a subderivation of d;*
(ii) *$d : B \vdash_\nu M : \sigma$ implies that every subterm of M, which is not under the scope of a λ-abstraction, is typed by a subderivation of d.*

The type systems $\vdash_*$ enjoy the subject-reduction property and a restricted form of subject-expansion.

Proposition 17 (Subject-reduction)
If $B \vdash_* M : \sigma$ and $M \to_\Lambda N$ then $B \vdash_* N : \sigma$.

PROOF. Standard, using the Generation Lemma.iv) and v). □

Proposition 18 (Typed subject-expansion)
Let $C[.]$ be a context. Then $B \vdash_* C[P[Q/x]] : \sigma$ and $B' \vdash_* Q : \tau$ imply $B \cap B' \vdash_* C[(\lambda x.P)Q] : \sigma$.

PROOF. We will show the proof for $\vdash_\nu$. The other proof is similar but simpler (see [6]).

The proof is by induction on $C[.]$. Let d be a derivation proving $B \vdash_\nu C[P[Q/x]] : \sigma$. We may assume, without loss of generality, that B is undefined on x and that all typings in d have the same basis B. Indeed, $(\to I)$ is the only rule having a basis, in the premises, different from the basis in the conclusion; but we can assume that free and bound variables have different names in M.
In case $C[.] = [.]$, there are two cases to analyze.

(i) Suppose that either $x \notin \mathrm{FV}(P)$ (hence $P[Q/x] \equiv P$) or Q occurs only in subterms of P which are subjects of an application of the rule (ν).
In both cases, $B \vdash_\nu P[Q/x] : \sigma$; therefore $B[\tau/x] \vdash_\nu P : \sigma$, by Lemma 15.ii), for every τ. Then $B \vdash_\nu \lambda x.P : \tau \to \sigma$, by rule $(\to I)$ and, by Lemma 15.i), both $B \cap B' \vdash_\nu \lambda x.P : \tau \to \sigma$ and $B \cap B' \vdash_\nu Q : \tau$. Hence, by rule $(\to E)$,

$$B \cap B' \vdash_\nu (\lambda x.P)Q : \sigma.$$

(ii) Suppose that Q occurs in $P[Q/x]$ and let $d_i : B \vdash_\nu Q : \sigma_i$ be the subderivation occurrences, being not inside subterms typed by the rule (ν), that we want to extract. The derivation d can be transformed into a derivation d' proving $B[\sigma_1 \wedge ... \wedge \sigma_n/x] \vdash_\nu P : \sigma$ by performing the following operations.

- Replace each typing $B \vdash_\nu Q : \sigma_i$ by:

$$\dfrac{\dfrac{}{B[\sigma_1 \wedge ... \wedge \sigma_n/x] \vdash_\nu x : \sigma_1 \wedge ... \wedge \sigma_n}\,(var)}{B[\sigma_1 \wedge ... \wedge \sigma_n/x] \vdash_\nu x : \sigma_i}\,(\wedge E_*)$$

  where $(\wedge E_*)$ denotes a sequence of applications of $(\wedge E_l)$ and $(\wedge E_r)$.
- Replace each occurrence of Q in $P[Q/x]$ by x.
- Replace each occurrence of B by $B[\sigma_1 \wedge ... \wedge \sigma_n/x]$.

It is easy to check that d' is well defined. By rule $(\rightarrow I)$ we obtain $B \vdash_\nu \lambda x.P : \sigma_1 \wedge ... \wedge \sigma_n \rightarrow \sigma$. Moreover, since Q is a subterm of $P[Q/x]$, then the free variables of Q are all in the domain of B, so there are derivations $B \vdash_\nu Q : \sigma_i$, and by repeatly applying rule $(\wedge I)$, we can build a proof of $B \vdash_\nu Q : \sigma_1 \wedge ... \wedge \sigma_n$, and the result follows by rule $(\rightarrow E)$.

For the general case, where $C[.] = \lambda x.C'[.]$ or $C[.] = C_1[.]C_2[.]$, the result follows easily by induction. $\Box$

5 Strong Normalization and Potentially valuability vs Typability

In this section, we will prove that both Λ-strong normalization and Λ-potential valuability imply typability in the system $\vdash$, and that both $\Lambda\ell$-strong normalization and $\Lambda\ell$-potential valuability imply typability in the system $\vdash_\nu$. Note that the result about Λ-strong normalization is already known, but we chosed to treat all cases by completeness. To this aim, let us recall the shape of a normal form in both the considered calculi. A term in Λ-normal form M is of the shape $\lambda x_1 \dots x_m.xM_1 \dots M_n$, where $m, n \geq 0$ and M_i is a Λ-normal form for all $1 \leq i \leq n$. A term in $\Lambda\ell$-normal form either is of the shape $\lambda x.M'$, for some $M' \in \Lambda$ or $xM_1 \dots M_n$ with $n \geq 0$ where M_i is a $\Lambda\ell$-normal forms for all $1 \leq i \leq n$.

Lemma 19 (i) *$M \in \Lambda$-NF implies it is typable in the system $\vdash$.*
(ii) *$M \in \Lambda\ell$-NF implies it is typable in the system $\vdash_\nu$.*

PROOF. In both cases the proof is carried out by induction on the structure of a normal form.

(i) Let $M \equiv \lambda x_1 \dots x_m.xM_1 \dots M_n$. If $n = 0$, and $m = 0$ the proof is trivial. Let $n > 0$ and $m = 0$. By inductive hypothesis there are $B_1, \dots, B_n$ and $\sigma_1, \dots, \sigma_n$ such that: $B_i \vdash M_i : \sigma_i$. Then M has type σ in the basis

$B' = B_1 \cap \ldots \cap B_n \cap [\sigma_1 \to \ldots \to \sigma_n \to \sigma / x]$ since:

$$\dfrac{\dfrac{}{B' \vdash x : \sigma_1 \to \ldots \to \sigma_n \to \sigma}\ (*) \quad \dfrac{\vdots}{B' \vdash M_i : \sigma_i}\ \ldots}{B' \vdash xM_1 \ldots M_n : \sigma}\ (\to E)$$

where $(*)$ denotes a sequence of applications of rules $((var))$, $(\wedge I)$, $(\wedge E_l)$ and $(\wedge E_r)$. In case $m > 0$, let $B'' = \{x_j : \tau_j \mid x_j \in \{x_1, ..., x_m\}, x_j \notin dom(B'), \tau_j fresh\}$. By Lemma 15.(ii), $B' \cap B'' \vdash xM_1 \ldots M_n : \sigma$, so $B''' \vdash \lambda x_1 \ldots x_m . xM_1 \ldots M_n : \rho_1 \to ... \to \rho_m \to \sigma$, where B''' is obtained from $B' \cap B''$ by erasing the assignments about x_i $(1 \leq i \leq m)$ and $x_i : \rho_i \in B' \cap B''$.

(ii) In case $M \equiv xM_1 \ldots M_n$ the proof is similar to the similar case of the previous point, just replacing $\vdash$ by $\vdash_\nu$. In the case $M \equiv \lambda x.M'$ then $B \vdash_\nu M : \nu$ for any basis B. $\square$

An innermost redex is a redex such that its argument is in normal form. It is easy to check that, if a term M is not in normal form, in any calculus, then it contains at least one innermost redex.

Theorem 20 (i) *$M \in \Lambda$-SN implies M is typable in $\vdash$.*
(ii) *$M \in \Lambda\ell$-SN implies M is typable in $\vdash_\nu$.*

PROOF.

(i) If M is in Λ-NF, then the proof follows from Lemma 19.(i). Otherwise, we can assume that there is a Λ-reduction sequence

$$M \equiv M_0 \to_\beta M_1 \to_\beta \ldots \to_\beta M_n \equiv N$$

reducing at each step the leftmost innermost redex ($n > 0$).This reduction sequence is finite, since M is Λ-strongly normalizing. The proof is given by induction on n.
By induction hypothesis, there are a base B_1 and a type σ such that $B_1 \vdash M_1 : \sigma$: let $(\lambda x.P)Q$ be the reduced redex. Then, there is a basis B_2 and a type τ such $B_2 \vdash Q : \tau$ by Lemma 19.(i). Then the result follows trivially from Property 18.(i).

(ii) Similar to the proof of the previous point, taking into account that the innermost redex cannot occur in a subterm typed by the type ν. $\square$

Corollary 21 (i) *$M \in \Phi$ implies M is typable in $\vdash$;*
(ii) *$M \in \Gamma$ implies M is typable in $\vdash_\nu$.*

PROOF.

(i) By Lemma 13.(iv) and Theorem 20.(i);
(ii) From Theorem 20.(i), observing that $\Gamma \subset \Lambda\ell$-NF.

Now let us consider the potential valuability property.

Theorem 22 (i) *$M \in \Phi$-PV implies M is typable in $\vdash$.*
(ii) *$M \in \Gamma$-PV implies M is typable in $\vdash_\nu$.*

PROOF.

(i) Let $M \in \Phi$-PV. Then there is a substitution $\mathbf{s}$, replacing variables by terms belonging to Φ, such that $\mathbf{s}(M) \to_\Phi^* N \in \Phi$. Then, by Lemma 21.(i), there are B and σ such that $B \vdash N : \sigma$. Moreover, since every Φ-reduction step is a β-reduction too, by Corollary 21 and Proposition 18 $B' \vdash \mathbf{s}(M) : \sigma$, for some B' such that $B \subseteq B'$. Let $FV(M) = \{x_1, \ldots, x_n\}$ $(n \geq 0)$, and let $\mathbf{s}(x_i) = P_i \in \Phi$. So, again by Corollary 21 and Proposition 18, $B'' \vdash (\lambda x_1 \ldots x_n.M)P_1 \ldots P_n : \sigma$, for some B'' such that $B' \subseteq B''$. Then, by Proposition 16, M is typable.
(ii) Let $M \in \Gamma$-PV. Then there is a substitution $\mathbf{s}$, replacing variables by terms belonging to Γ, such that $\mathbf{s}(M) \to_\Gamma^* N \in \Gamma$. Let $FV(M) = \{x_1, \ldots, x_n\}$ $(n \geq 0)$, and let $\mathbf{s}(x_i) = P_i \in \Gamma$. By mimicking the proof of the previous point, we can obtain that $d : B \vdash_\nu (\lambda x_1 \ldots x_n.M)P_1 \ldots P_n : \sigma$, for some B and σ. In order to conclude, assume M is not typed by a subderivation of d. But in this case it there must be j such that $d' : B' \vdash_\nu \lambda x_j \ldots x_n.M : \nu$ is a subderivation of d $(1 \leq j \leq n)$. But, since the type ν has not applicative power, only the subterm $(\lambda x_1 ... \lambda x_{j-1}.M)P_1 \ldots P_{j-1}$ can be typed, contrary to what we obtained before. □

6 Typability vs strong normalization and potential valuability

One of the tools used in the literature for proving the strong normalization of a type assignment system is reducibility, introduced by Tait [19]. Here we need a stronger result, since we want to prove that typability implies both strong normalization and potential valuability. In order to prove both implications at the same time, we use a reducibility method, based on a non standard definition of saturated sets.

Definition 23 (i) *Let a k-saturated set S_k be a set such that:*
1. *$S_k \subseteq \Lambda$-SN;*
2. *$x \in$ Var and $M_i \in \Lambda$-SN imply $xM_1 ... M_n \in S_k$ $(1 \leq i \leq n)$;*
3. *$M[P/x]M_1 ... M_n \in S_k$ and $P \in \Lambda$-SN imply $(\lambda x.M)PM_1 ... M_n \in S_k$.*

4. $\forall h \geq k$, $O^h \in S_k$, *where* $O^h \equiv \lambda x_1 \ldots x_h x_{h+1}.x_{h+1}$.

Let SAT_k *be the set of all* k*-saturated sets.*

(ii) *A* k*-lazy saturated set* S_k^ℓ *is defined in a similar way as a* k*-saturated set, only replacing* Λ*-SN by* $\Lambda\ell$*-SN. Let* SAT_k^ℓ *be the set of all* k*-lazy saturated sets.*

(iii) $SAT = \bigcup_{k\in\omega} SAT_k$ *and* $SAT^\ell = \bigcup_{k\in\omega} SAT_k^\ell$.

We will call saturated set (lazy-saturated set) a k-saturated set (k-lazy saturated set), for some k. Note that the previous definition differs from the classical one by adding one more condition (item 4). Hence there are saturated sets according to the classical definition which are not saturated in our sense (e.g. the least saturated set containing all variables).

Theorem 24 (i) Λ*-SN* $\subseteq SAT$.
(ii) $\Lambda\ell$*-SN* $\subseteq SAT^\ell$.

PROOF. In both cases, the proof is obvious. □

Let $\mathtt{S}$ and $\mathtt{T}$ be either two saturated sets or lazy-saturated sets. Define: $\mathtt{S} \to \mathtt{T} = \{M \mid MN \in \mathtt{T}, \text{for all } N \in \mathtt{S}\}$. In order to prove that $\mathtt{S} \to \mathtt{T}$ is saturated, we need a further property.

Proposition 25 (i) $\mathtt{T} \in SAT_k$ *implies that* $O^{k+1}N \in \mathtt{T}$*, for all* $N \in \Lambda$*-SN.*
(ii) $\mathtt{T} \in SAT_k^\ell$ *implies that* $O^{k+1}N \in \mathtt{T}$*, for all* $N \in \Lambda\ell$*-SN.*

PROOF. Both proofs follow from Definition 23.(iii). □

Lemma 26 (i) $\mathtt{T} \in SAT_t$ *implies that* $(\mathtt{S} \to \mathtt{T}) \in SAT_{t+1}$*, for all* $\mathtt{S} \in SAT$.
(ii) $\mathtt{T} \in SAT_t^\ell$ *implies that* $(\mathtt{S} \to \mathtt{T}) \in SAT_{t+1}^\ell$*, for all* $\mathtt{S} \in SAT^\ell$.

PROOF. Both proofs follow trivially from Definition of $\to$ and Proposition 25, since $\mathtt{S}$ is respectively included in Λ-SN and $\Lambda\ell$-SN. □

Now we will interpret types as (lazy) saturated sets, and bases as sets of substitutions, in the following way.

Definition 27 (i) *If* $\rho : \mathrm{Var} \to SAT$ *then* $[\![.]\!]_\rho$ *is the function from types to saturated set defined as follows,*

- $[\![\alpha]\!]_\rho = \rho(\alpha)$;
- $[\![\sigma \to \tau]\!]_\rho = [\![\sigma]\!]_\rho \to [\![\tau]\!]_\rho$;
- $[\![\sigma \wedge \tau]\!]_\rho = [\![\sigma]\!]_\rho \cap [\![\tau]\!]_\rho$.

(ii) *If* $\rho : \mathrm{Var} \to SAT^\ell$ *then* $[\![.]\!]^\ell_\rho$ *is the function from types to lazy saturated set defined as follows,*
- $[\![\alpha]\!]^\ell_\rho = \rho(\alpha)$*;*
- $[\![\nu]\!]^\ell_\rho = \Lambda\ell$*-SN;*
- $[\![\sigma \to \tau]\!]^\ell_\rho = [\![\sigma]\!]^\ell_\rho \to [\![\tau]\!]^\ell_\rho$*;*
- $[\![\sigma \wedge \tau]\!]^\ell_\rho = [\![\sigma]\!]^\ell_\rho \cap [\![\tau]\!]^\ell_\rho$.

(iii) *If* $B = x_1 : \sigma_1, ..., x_n : \sigma_n$ *then* $[\![B]\!]_\rho = \{\mathbf{s} \mid \mathbf{s}(x_i) \in [\![\sigma_i]\!]_\rho\}$ *and* $[\![B]\!]^\ell_\rho = \{\mathbf{s} \mid \mathbf{s}(x_i) \in [\![\sigma_i]\!]^\ell_\rho\}$.

Both the type assignment systems are correct with respect to the previous defined semantics.

Lemma 28 (i) $B \vdash M : \sigma$ *implies* $\forall \rho : \mathrm{Var} \to SAT$, $\forall \mathbf{s} \in [\![B]\!]_\rho$, $\mathbf{s}(M) \in [\![\sigma]\!]_\rho$.

(ii) $B \vdash_\nu M : \sigma$ *implies* $\forall \rho : \mathrm{Var} \to SAT^\ell$, $\forall \mathbf{s} \in [\![B]\!]^\ell_\rho$, $\mathbf{s}(M) \in [\![\sigma]\!]^\ell_\rho$.

PROOF.

(i) By induction on the derivation. If the last applied rule is (var) then the result is obvious. In the case the last applied rule is

$$\frac{B[\sigma/x] \vdash M : \tau}{B \vdash \lambda x.M : \sigma \to \tau}\ (\to I)$$

by induction, $\forall \rho, \forall \mathbf{s} \in [\![B]\!]_\rho$, $\mathbf{s}(M) \in [\![\tau]\!]_\rho$. Since $\mathbf{s}(x) \in [\![\sigma]\!]_\rho$, the result follows by the definition of $[\![\sigma]\!]_\rho \to [\![\tau]\!]_\rho$. In case the last applied rule is

$$\frac{B \vdash M : \sigma \to \tau \quad B \vdash N : \sigma}{B \vdash MN : \tau}\ (\to E)$$

by induction, $\forall \rho . \forall \mathbf{s} . \mathbf{s}(M) \in [\![\sigma \to \tau]\!]_\rho$ and $\mathbf{s}(N) \in [\![\sigma]\!]_\rho$. Then $\mathbf{s}(MN) \in [\![\tau]\!]_\rho$, by definition of $[\![\sigma]\!]_\rho \to [\![\tau]\!]_\rho$. The cases dealing with the rules involving $\wedge$ come immediately by induction.

(ii) (ν) is obvious. Further cases are similar to that of the previous point. □

Proposition 29 *Let* $\mathbf{O}$ *be the set of all substitutions mapping each variable* x_i *to a term of the shape* O^{k_i}*, for some* $k_i \in \mathbb{N}$.

(i) $B \vdash M : \sigma$ *implies that there exists* $\mathbf{o} \in \mathbf{O}$ *such that* $\mathbf{o}(M) \in \Lambda$*-SN.*

(ii) $B \vdash_\nu M : \sigma$ *implies that there exists* $\mathbf{o} \in \mathbf{O}$ *such that* $\mathbf{o}(M) \in \Lambda\ell$*-SN.*

PROOF.

(i) Let $B = x_1 : \tau_1, ..., x_n : \tau_n$, and let $\rho : \mathrm{Var} \to SAT$. Definition 27 implies $O^{k_i} \in [\![\tau_i]\!]_\rho$, for some $k_i \in \mathbb{N}$. Let $\mathbf{o}(x_i) = O^{k_i}$, so $\mathbf{o} \in [\![B]\!]_\rho$. Then by Lemma 28 $M \in [\![\sigma]\!]_\rho$, so $M \in \Lambda$-SN.
(ii) Similar to that of the previous case. □

Lemma 30 (i) *$M \in \Lambda$-SN implies $M \to_\Phi^* N \in \Upsilon$.*
(ii) *$M \in \Lambda\ell$-SN and $M \in \Lambda^0$ implies $M \to_\Gamma^* N \in \Gamma$*

PROOF.

(i) To each $M \in \Lambda$-SN, associate the number $l(M)$ that is the maximum length of a derivation $M \to^* N$, where $N \in \Lambda$-NF. Note that, if $M \to_\Lambda N$ then there is a sequence of reductions starting from M of length $1+l(N)$, hence $l(N)$ is strictly less than $l(M)$.
The proof is by induction on the pair $(l(M), |M|)$. If $M \equiv \lambda x.P$ or $M \equiv xM_1 \dots M_m$ the result follows easily by induction.
Let $M \equiv (\lambda x.P)QM_1 \dots M_m$. If M contains at least one Φ-redex, then the proof follows by induction. Otherwise, by induction $Q, M_1, \dots, M_m \in \Upsilon$. Moreover, $P[Q/x]M_1 \dots M_m \to_\Phi^* N \in \Upsilon$ by induction. Note that Q is neither a variable nor a closed term, otherwise $(\lambda x.P)Q$ would be a Φ-redex. Therefore, $M \in \Upsilon$ by the definition of Υ.
(ii) The proof is easy by induction on $|M|$. □

Theorem 31 (i) *$B \vdash M : \sigma$ implies $M \in \Phi$-PV.*
(ii) *$B \vdash_\nu M : \sigma$ implies $M \in \Gamma$-PV.*

PROOF.

(i) By Property 29.(i) and by Lemma 30.(i).
(ii) By Property 29.(ii) and by Lemma 30.(ii). □

Theorem 32 (i) *$B \vdash M : \sigma$ implies $M \in \Lambda$-SN.*
(ii) *$B \vdash_\nu M : \sigma$ implies $M \in \Lambda\ell$-SN.*

PROOF.

(i) Let $B = x_1 : \tau_1, ..., x_n : \tau_n$, and let $\rho : \mathrm{Var} \to SAT$. Then $x_i \in [\![\tau_i]\!]_\rho$ (by Definition 27.(iii)). Let $\mathbf{s}$ be such that $\mathbf{s}(x_i) = x_i$, so $\mathbf{s} \in [\![B]\!]_\rho$. Then, by Lemma 28, $M \in [\![\sigma]\!]_\rho$, so $M \in \Lambda$-SN.
(ii) Similar to that of the previous case. □

7 On the quality of the proposed solutions

In Section 3 we refined the two problems of characterizing the notion of (lazy) strongly-normalization through potential valuability, asking also for the existence of both decidable and minimal solutions. In this last section we will show that the solutions we propose are good, in some sense. In particular, we will prove that the set Φ is a minimal solution of Problem 1. Clearly Φ is a non recursive set, but we will prove that a decidable solution does not exist. Moreover we will show that Γ, although decidable, is not a minimal solution of Problem 2, but a minimal solution does not exist.

Theorem 33 *Φ is minimal between all the solutions of Problem 1.*

PROOF. Let $\Delta^\star$ be a set of input values.
We will prove that, if the set of potentially $\Delta^\star$-valuable terms coincides with the set of the strongly Λ-normalizing terms and $\Delta^\star \subset \Phi$ then $\Delta^\star = \Phi$. Clearly $\Delta^\star = \Phi$ if and only if $(\Delta^\star)^0 = \Phi^0$, since $\Delta^\star \subset \Phi$ and $\Phi = \mathrm{Var} \cup \Phi^0$. Let $M \in \Phi^0$. Note that M is Φ-valuable, potentially Φ-valuable, in Φ-normal form and also a closed strongly Λ-normalizing term. Thus, M is potentially $\Delta^\star$-valuable by hypothesis and $M \in \Lambda^0$ implies that M is $\Delta^\star$-valuable. But $\Delta^\star \subset \Phi$ implies Φ-NF $\subset \Delta^\star$-NF, hence $M \in \Delta^\star$-NF. Then $M \in \Delta^\star$ and the proof is done. $\Box$

Also, it is worthy to say that, although Φ is minimal, it is not *the* minimum set answering Problem 1. In fact, the minimum solution to the following equations:

$$\begin{aligned}\Theta = &\{\lambda x_0 ... x_n.y \mid y \neq x_i \ \ (0 \leq i \leq n)\ \} \cup \\ &\{xM_1...M_n \mid M_k \in \Theta\ (1 \leq k \leq n)\} \cup \{\lambda \vec{x}.M \mid M \in \Theta\} \cup \\ &\{(\lambda x.P)QM_1...M_n \mid Q, M_1, ..., M_n \in \Theta,\ Q \notin \Delta,\ P[Q/x]M_1...M_n \to_\Delta^* R \in \Theta\}\end{aligned}$$

$$\Delta = \{\lambda x_0 ... x_n.y \mid y \neq x_i \ \ (0 \leq i \leq n)\ \} \cup (\Theta)^0$$

is also a solution to Problem 1. Sets Θ and Φ are not comparable, in fact $\lambda x.y \in \Theta$ but not in Φ, while $I(\lambda x.y) \in \Phi$ but not to Θ.

In order to prove the next result, we need to recall a property, first proved in [10].

Proposition 34 *For every term M, there is an effective procedure building two Λ-normal forms, P_M and Q_M, such that $P_M Q_M \to_\Lambda^* M$.*

PROOF. The proof is by induction on the structure of M. If $M \equiv x$, then $P_x Q_x \equiv (\lambda y.y)x$. If $M \equiv \lambda x.N$, then by induction there are P_N and Q_N such

that $P_N Q_N \to_\Lambda^* N$. So $P_M Q_M \equiv (\lambda x.xQ_N)P_N$, where x is fresh. If $M \equiv NR$, then $P_M Q_M \equiv (\lambda y.yP_N Q_N(yP_R Q_R))I$, where y is fresh.

Theorem 35 *There isn't a decidable set of input values which is a solution of Problem 1.*

PROOF. Assume that such a set, say $\Delta^\#$, exists. Since $D \equiv \lambda x.xx$ is a closed Λ-normal form, it closed $\Delta^\#$-normal form, and hence it must be in $\Delta^\#$. Henceforth, xx cannot belong to $\Delta^\#$, since sets of input values need to be closed under substitution and DD is not a strongly normalizing term. Let $M \equiv (\lambda z.P)(xx)Q$ where $P, Q \in \Lambda\text{-NF}^0$. M is an open $\Delta^\#$-normal form, so it does not necessarily belong to $\Delta^\#$.
Clearly, M is Λ-strongly normalizing if and only if PQ is Λ-strongly normalizing. Let $\mathbf{s}$ be a substitution replacing x by I. $\mathbf{s}(M) \to_{\Delta^\#}^* PQ$, since I belongs to every set of input values (satisfying the Problem 1). Therefore M belongs to $\Delta^\#$ if and only if PQ is Λ-strongly normalizing. So the problem is reduced to that one of deciding if a term which is an application of two Λ-normal forms is Λ-strongly normalizing. But, by the previous property, this problem is equivalent to the general Λ-strongly normalization problem, which is well known to be undecidable (see [20]). □

Hence, it is reasonable to say that Φ is *the best* solution to Problem 1, since it is minimal (but the minimum does not exist) and semi-decidable (while there cannot be a decidable one).

About Problem 2, it is interesting to note that Γ is not a minimal solution of it. Indeed, $\Gamma_D = \Gamma - \{M \in \Gamma | M \to_\Gamma^* \lambda x.xx\}$ is a proper subset of Γ, and it is also a solution to Problem 2. This remark is the starting point for proving that no minimal solution exists.

Theorem 36 *There is not a minimal solution of Problem 2.*

PROOF. Let Δ be a solution of Problem 2. Let $\omega_3^k \equiv \lambda z.\overbrace{(\lambda x.xxx)...(\lambda x.xxx)}^{k+2}$ for all $k \in \mathbb{N}$. Since $D_3 \equiv \lambda x.xxx$ is a closed normal form, then $D_3 \in \Delta$ by hypothesis. Note that ω_3^k contains a single redex and $\omega_3^k \to_\Delta \omega_3^{k+1}$ for all $k \in \mathbb{N}$, since $D_3 \in \Delta$.
Each ω_3^k must be Δ-valuable, since $\omega_3^k \in \Lambda\ell\text{-NF}^0$; thus, there exists $n \in \mathbb{N}$ such that $\omega_3^n \in \Delta$. Indeed Δ contains an infinite subset of ω_3^k-terms.
Clearly $\Delta^* = \Delta - \{\omega_3^n | n \in \mathbb{N}\}$ is strictly contained in Δ, but it is again a set of input values such that its potentially valuables terms correspond exactly to that of $\Lambda\ell$-strongly normalizing terms. □

Hence we could say that Γ is the best solution to Problem 2, since it is decidable, with an easy syntax, although not minimal, but a minimal solution does not exist.

References

[1] Abramsky S., Ong L.C., "Full abstraction in the Lazy Lambda Calculus", *Information and Computation*, 105, 1993, pp. 159-267.

[2] van Bakel S., *Intersection type assignment systems*, Theoretical Computer Science, 38(2):246-269, Elsevier, 1997.

[3] Barendregt H.P., *The Lambda Calculus: its syntax and semantics*, N.103 in Studies in Logic and the Foundations of Mathematics (revised edition), North-Holland, Amsterdam, 1994.

[4] Böhm, C. Alcune proprietà delle forme $\beta\eta$-normali nel λK-calcolo, Pubblicazioni dell'istituto per le applicazioni nel calcolo, n. 696, Roma, 1968.

[5] Coppo M., Dezani-Ciancaglini M., Zacchi M., *Type Theories, Normal Forms and $\mathcal{D}_\infty$ Lambda Models*, Information and Control, 72, 2, 1987, pp.85-116.

[6] Coppo M., Dezani-Ciancaglini M., *An Extension of the Basic Functionality Theory for the λ-Calculus* Notre-Dame Journal of Formal Logic, 21(4), pp. 685-693, October 1980.

[7] Egidi L., Honsell F., and Ronchi della Rocca S., *Operational, denotational and logical descriptions: a case study,* Fundamenta Informaticae, 16(2) 149–170, 1992.

[8] Krivine J.L., *Lambda-Calculus, Types and Models*, Ellis Horwood Series in Computers and Their Applications. 1993.

[9] Landin P.J., *The mechanical evaluation of expressions*, Computer Journal, 1964.

[10] Nederpelt R., *Strong normalisation in a lambda calculus with lambda structured types*, PhD thesis, Eindhoven University of Technology, The Netherlands, 1973.

[11] Paolini L.,*Call-by-value separability and computability*, ICTCS'01, Restivo, Ronchi Della Rocca, and Roversi, eds, LNCS 2202, Springer-Verlag, 74-89.

[12] Paolini L., Pimentel E., Ronchi Della Rocca S. *Lazy strong normalization*, ITRS'04, ENTCS vol. 136, pp. 103–116, 2005.

[13] Paolini L., Pimentel E., Ronchi Della Rocca S. *An Operational characterization of Strong Normalization*, FOSSACS'06, LNCS vol. 3835, pp. 352–366, 2006.

[14] Paolini L., Ronchi Della Rocca S., *Call-by-value Solvability*, Theoretical Informatics and Applications, 33(6), 507-534, 1999.

[15] Paolini L., Ronchi Della Rocca S., *The Parametric Parameter Passing λ-calculus*, Information and Computation, 189(1):87-106, 2004.

[16] Plotkin G.D., *Call-by-name, call-by-value and the λ-calculus*, Theoretical Computer Science (1) 125-159, 1975.

[17] Pottinger G., *A type assignment for the strongly normalizable λ-terms*, in To H.B. Curry: essays on combinatory logic, lambda calculus and formalism, pp.561-577, Academic Press, London, 1980.

[18] Ronchi Della Rocca S., Paolini L., *The Parametric λ-calculus. A meta-model for computation*, Texts in Theoretical Computer Science: an EATCS Series, Springer-Verlag, Berlin, 2004.

[19] Tait, W. W., *Intentional interpretations of functionals of finite type I* Journal of Symbolic Logic, 32(2):198-212, 1967.

[20] Urzyczyn, P., *A simple proof of undecidability of Strong Normalization*, Notes, 2000.