Solvability for Generalized Applications

Delia Kesner
Université de Paris, CNRS, IRIF, France
Institut Universitaire de France, France

Loïc Peyrot
Université de Paris, CNRS, IRIF, France

Abstract

Solvability is a key notion in the theory of call-by-name λ-calculus, used in particular to identify meaningful terms. However, adapting this notion to other call-by-name calculi, or extending it to different models of computation – such as call-by-value – , is not straightforward. In this paper, we study solvability for call-by-name and call-by-value λ-calculi with generalized applications, both variants inspired from von Plato's natural deduction with generalized elimination rules. We develop an operational as well as a logical theory of solvability for each of them. The operational characterization relies on a notion of solvable reduction for generalized applications, and the logical characterization is given in terms of typability in an appropriate non-idempotent intersection type system. Finally, we show that solvability in generalized applications and solvability in the λ-calculus are equivalent notions.

2012 ACM Subject Classification Theory of computation → Lambda calculus; Theory of computation → Rewrite systems; Theory of computation → Type theory

Keywords and phrases Lambda-calculus, Generalized applications, Solvability, CBN/CBV, Quantitative types

Digital Object Identifier 10.4230/LIPIcs.FSCD.2022.18

1 Introduction

The λ-calculus with generalized applications ΛJ [24, 25] can be seen as a Curry-Howard interpretation of natural deduction with generalized elimination rules, a system developed in parallel by Tenant [34] and von Plato [35]. Like the usual λ-calculus, ΛJ is a call-by-name (CBN) calculus: the arguments of a function are substituted in its body without any other prior computation. In the call-by-value (CBV) paradigm instead, the arguments are always reduced before substitution [33]. CBV parameter passing is a choice made by many functional programming languages such as OCaml, Scheme or Coq. Very recently, a CBV variant of ΛJ, called ΛJ_v, was introduced by Espírito Santo [16]. It is surprisingly simple, implements general *strong* reduction (inside abstractions), and is very close to the CBN formalism ΛJ: they both share the same notion of reducible expression (*a.k.a. redex*), so that *every* function application is always reducible, because only the underlying notion of substitution differs.

The study of λ-calculi with generalized applications is of special interest for the understanding of the semantics of programming language. They give a fresh look at applications, and also bear similarities with explicit substitutions [1, 27] and the sequent calculus [19]. However, while many previous works [15, 17] were dedicated to the proof-theory underlying different calculi with generalized applications, very few semantical properties have been studied so far in this framework. In this paper, we show that existing tools and techniques for *solvability* can be adapted to this more general model of computation.

Solvability is used to identify *meaningful* terms. Indeed, all solvable terms progressively unveil a stable structure along the reduction process: this gives a step-by-step partial result that is later integrated into the definitive structure of the fully normalized term. Thus, the set of solvable terms contains all normalizable terms, but also a strict subset of the divergent ones. On the contrary, if a term containing an unsolvable subterm u converges to a result, then u can be replaced by any other term, still giving the same result and thus justifying the designation of unsolvable as *meaningless* (Genericity Lemma [8]). In semantical models of the λ-calculus, equating all unsolvable terms (*e.g.* to bottom) turns out to be consistent, but this is not the case if one equates in addition terms which are not normalizing [36].

Whilst being an important semantical property, solvability also has a very elegant operational theory. A solvable term may reduce to any other term when closed by abstractions and applied to a suitable sequence of arguments (*i.e.* plugged inside a *head context*). In the CBN λ-calculus, a term t is solvable *iff* t has a *head normal form iff* t *head*-normalizes [36]. But because of the different normalization behaviors of CBN/CBV, their corresponding notions of solvability do not perfectly coincide. In fact, it is only recently that a characterization of CBV solvability making use of some proper notion of CBV reduction has been achieved [7, 12].

In this paper, we give an *operational* characterization of solvability for CBN/CBV generalized applications in terms of an appropriate reduction relation for each case. Indeed, we define a notion of *solvable reduction* for CBN (resp. CBV) for which all and only CBN (resp. CBV) solvable terms normalize. We take advantage of the similarity between CBN and CBV with generalized applications (given by the fact that they both share the same notion of redex, in contrast to the λ-calculus case) to highlight the divergences of the two kinds of solvability. On the one hand, our results show the robustness of the operational theory of solvability. On the other hand, our study of solvability is a step forward in the understanding of programming language semantics based on generalized applications.

In addition to the previously mentioned operational characterizations, we also provide *logical* characterizations of CBN/CBV solvability as typability in *quantitative* (*a.k.a.* non-idempotent intersection) *type systems.* More precisely, we deduce an equivalence between (1) being normalizable for the newly defined solvable reduction relations, (2) being (quantitatively) typable and (3) being solvable. In the λ-calculus, such a result was already well-known for CBN (see [11] for a survey), and it has recently been obtained [5] for CBV, serving as the starting point for our CBV characterization. We will then also prove equivalence between solvability in the CBN/CBV λ-calculus and solvability in CBN/CBV generalized applications by simply reasoning on their associated quantitative type systems. A further advantage of quantitative types is that they enable simple combinatorial proofs of normalization. They also have strong ties with denotational semantics, as it is generally possible to derive a relational model from a quantitative type system.

The adaptation of CBN/CBV solvability in our framework is not trivial, because of the particular syntactical structure of generalized applications which facilitates the existence of *blocked* redexes. To this end, the original calculi with generalized applications ΛJ and ΛJ_v use a permutative/commutative conversion π on top of the primary rule β for CBN and $\beta_{\mathtt{v}}$ for CBV, respectively. This rule is needed to unblock these stuck redexes. Instead, we work with *distant* variants (λJ for CBN [18], written here λJ_n, and a novel distant variant λJ_v for CBV). Thus, we integrate the permutations into the primary rules, in the spirit of calculi with explicit substitutions at a distance [6]. In this way, the reduction rules focus on the computational behavior of the language and not on syntactical accidents. In particular, only primary reductions (and not permutations) create divergence, and thus unsolvability. This choice also reflects the logical model in a better way: in our framework, quantitative type systems are neutral to permutations rules (specifically: the size of derivations does not change), as quantitativity is only related to the computational rules. Yet, we show that our results also apply to the original calculi ΛJ and ΛJ_v.

To summarize, our main contributions are the following:

- We define appropriate notions of head contexts and solvable reductions in calculi with generalized applications and distance.
- We show operational characterizations of CBN/CBV solvability in this setting.
- We also provide (logical) quantitative characterizations of these notions of solvability.
- We derive characterizations of solvability for the original calculi with generalized applications (without distance).
- We show that our definition of solvability corresponds to the one of the λ-calculus.

Plan of the paper

In Sec. 2 we present the distant calculi with generalized applications for both CBN and CBV. We then introduce their respective notions of solvability in Sec. 3. We give operational and logical characterizations of CBN (resp. CBV) solvability in Sec. 4 (resp. Sec. 5). In Sec. 6 we extend our results to the original (non-distant) calculi ΛJ and ΛJ_v and we relate these results to solvability in the λ-calculus. Related works and conclusions are detailed in Sec. 7.

2 The Distant Calculi with Generalized Applications

In this section we define the syntax and operational semantics of two distant λ-calculi with generalized applications for call-by-name and call-by-value, noted λJ_n and λJ_v respectively.

2.1 Syntax

Given a countably infinite set $\mathcal{X}$ of variables $x, y, z \ldots$, the set of terms generated by the following grammar is called $\mathtt{T}_J$.

$$(\textbf{Values})\ v ::= x \in \mathcal{X} \mid \lambda x.t \qquad (\textbf{Terms})\ t,u,r,s ::= v \mid t(u, x.r)$$

From now on, values are denoted by v and arbitrary terms by t, u, r or s. The term x is a **variable**, $\lambda x.t$ is an **abstraction** and $t(u, x.r)$ is a **generalized application**, whose part $x.r$ – that is not a subterm itself – is called a **continuation**. A generalized application can be seen as a let-binding/explicit substitution under the (informal) translation of $t(u, x.r)$ to `let` x `=` tu `in` r. **Free** and **bound** variables of terms are defined as expected, in particular, $\mathrm{fv}(\lambda x.r) := \mathrm{fv}(r) \setminus \{x\}$ and $\mathrm{fv}(t(u, x.r)) := \mathrm{fv}(t) \cup \mathrm{fv}(u) \cup (\mathrm{fv}(r) \setminus \{x\})$, so that both $\lambda x.r$ and $t(u, x.r)$ bind the variable x in the term r. We work modulo α-conversion, so that bound variables can be renamed when necessary. A generalized application $t(u, x.r)$ is said to be **non-relevant** if $x \notin \mathrm{fv}(r)$.

We set some special terms that will be used in forthcoming examples and definitions: $\mathtt{I} := \lambda z.z$, $\delta := \lambda x.x(x, z.z)$, $\Omega := \delta(\delta, z.z)$, as well as a family of projection terms $\mathbf{o}^n := \lambda x_n \ldots \lambda x_0.x_0$ parametrized by a natural number n.

Contexts ($\mathtt{C}$) are terms with one occurrence of the hole $\Diamond$, and **distant contexts** ($\mathtt{D}$) are special contexts used to define the operational semantics of the calculi:

$$\mathtt{C} ::= \Diamond \mid \lambda x.\mathtt{C} \mid \mathtt{C}(u, x.r) \mid t(\mathtt{C}, x.r) \mid t(u, x.\mathtt{C}) \qquad \mathtt{D} ::= \Diamond \mid t(u, x.\mathtt{D})$$

The term $\mathtt{C}\langle t\rangle$ is obtained by replacing $\Diamond$ in the context $\mathtt{C}$ by the term t, so that capture of variables may eventually occur. When the free variables of t are not captured by $\mathtt{C}$, we may write $\mathtt{C}\langle\langle t\rangle\rangle$ instead of $\mathtt{C}\langle t\rangle$. Notice that every term can be (uniquely) decomposed into $\mathtt{D}\langle v\rangle$, where $\mathtt{D}$ is a distant context and v a value. Thus for example, let $t := x_1(u, y.x_2(y, z.z))$. Then, there are three possible decompositions of t in terms of distance contexts: $t = \mathtt{D}_0\langle x_1(u, y.x_2(y, z.z))\rangle$ with $\mathtt{D}_0 = \Diamond$, $t = \mathtt{D}_1\langle x_2(y, z.z)\rangle$ with $\mathtt{D}_1 = x_1(u, y.\Diamond)$ and $t = \mathtt{D}_2\langle z\rangle$ with $\mathtt{D}_2 = x_1(u, y.x_2(y, z.\Diamond))$.

2.2 CBN and CBV Operational Semantics

The CBN operational semantics relies on a notion of **right substitution**, which is the expected capture-free meta-level substitution on $\mathtt{T}_J$-terms:.

$$\begin{array}{rrcl@{\qquad}rcl} & \{u/x\}x & := & u & \{u/x\}(\lambda y.t) & := & \lambda y.\{u/x\}t \\ (x \neq y) & \{u/x\}y & := & y & \{u/x\}(t(s,y.r)) & := & (\{u/x\}t)(\{u/x\}s, y.\{u/x\}r) \end{array}$$

The CBV operational semantics is based on **left substitution**:

$$\{v\backslash\!\!\backslash x\}t \;:=\; \{v/x\}t \qquad\qquad \{s(u,y.r)\backslash\!\!\backslash x\}t \;:=\; s(u,y.\{r\backslash\!\!\backslash x\}t)$$

Notice however that left substitution of a value invokes the right substitution, and left substitution of a generalized application performs a commutative/permutative conversion. Thus for example, $\{\mathtt{I}\backslash\!\!\backslash x\}(x(\mathtt{I},y.y)) = \{\mathtt{I}/x\}(x(\mathtt{I},y.y)) = \mathtt{I}(\mathtt{I},y.y)$ and $\{\mathtt{I}(\mathtt{I},y.y)\backslash\!\!\backslash z'\}z = \mathtt{I}(\mathtt{I},y.\{y\backslash\!\!\backslash z'\}z) = \mathtt{I}(\mathtt{I},y.\{y/z'\}z) = \mathtt{I}(\mathtt{I},y.z)$. Given the unique decomposition of a term u into $\mathtt{D}\langle v\rangle$, we can alternatively define left substitution as: $\{u\backslash\!\!\backslash x\}t = \{\mathtt{D}\langle v\rangle\backslash\!\!\backslash x\}t := \mathtt{D}\langle\{v/x\}t\rangle$. This alternative definition highlights the principle of CBV, which consists of only substituting values. Thus, in the last example, although z' is different from z, the context $\mathtt{I}(\mathtt{I},y.\Diamond)$ is not erased, as it is pushed out of the substitution. Notice that $\{u\backslash\!\!\backslash x\}v$ is a value if and only if u is a value. Notice also that $\{t\backslash\!\!\backslash x\}x = t$.

Before giving the reduction rules of our calculi, we first define some general notations. We denote a **reduction rule** $\mathtt{r} \subseteq \mathtt{T}_J \times \mathtt{T}_J$ as $\mapsto_{\mathtt{r}}$. In this paper, **reduction relations**, denoted by $\to_{\mathcal{R}}$, are generated by a finite set of reduction rules closed under some particular contexts. Given a reduction relation $\to_{\mathcal{R}}$, we write $\to^*_{\mathcal{R}}$ (resp. $\to^+_{\mathcal{R}}$) for the reflexive-transitive (resp. transitive) closure of $\to_{\mathcal{R}}$. A term t is said to be in **$\mathcal{R}$-normal form** (written $\mathcal{R}$-nf) iff there is no t' such that $t \to_{\mathcal{R}} t'$.

The reduction relation $\to_{\mathtt{n}}$ of the original CBN calculus ΛJ_n [24, 25] is defined as the closure under all contexts of the following two reduction rules β and π. The reduction relation $\to_{\mathtt{v}}$ of the original CBV calculus ΛJ_v [16] is defined as the closure under all contexts of rules $\beta_{\mathtt{v}}$ and π.

$$\begin{array}{rrcl} (\textbf{CBN } \boldsymbol{\beta}\textbf{-Rule}) & (\lambda x.t)(u,y.r) & \mapsto_\beta & \{\{u/x\}t/y\}r \\ (\textbf{CBV } \boldsymbol{\beta}_{\mathtt{v}}\textbf{-Rule}) & (\lambda x.t)(u,y.r) & \mapsto_{\beta_{\mathtt{v}}} & \{\{u\backslash\!\!\backslash x\}t\backslash\!\!\backslash y\}r \\ (\textbf{Permutative } \boldsymbol{\pi}\textbf{-Rule}) & t(u,x.r)(u',y.r') & \mapsto_\pi & t(u,x.r(u',y.r')) \end{array}$$

We use instead a *distant* operational semantics, where permutations are directly integrated in the primary CBN/CBV rule of the calculus. We do not dispense with permutations altogether, as they are necessary to unblock $\beta/\beta_{\mathtt{v}}$-redexes when the function is not exactly next to its argument. This way, the distant paradigm focuses on the *computational* content of the calculus by using demand-driven permutations, only when primary redexes are blocked.

▶ **Definition 1.** *The **CBN calculus** λJ_n is defined by the grammar $\mathtt{T}_J$ equipped with the **distant call-by-name** reduction relation $\to_{\mathtt{dn}}$ [18]. The relation $\to_{\mathtt{dn}}$ is generated by the closure under all contexts $\mathtt{C}$ of the reduction rule: $\mathtt{D}\langle\lambda x.t\rangle(u,y.r) \mapsto_{\mathtt{d}\beta} \{\mathtt{D}\langle\{u/x\}t\rangle/y\}r$.*

An intuitive explanation of this rule can be given through the previous informal translation of generalized applications to let-bindings: $\mathtt{D}\langle\lambda x.t\rangle(u,y.r)$ corresponds to `let` y `=` $\mathtt{D}\langle\lambda x.t\rangle u$ `in` r. In this first term, the computation in the foreground comes from the function $\mathtt{D}\langle\lambda x.t\rangle$ and its argument u. We get an intermediate result by substituting u for x in t inside the distant context $\mathtt{D}$, thus obtaining `let` y `=` $\mathtt{D}\langle\{u/x\}t\rangle$ `in` r. This intermediate result can then be fed to the continuation by unfolding the let-binding, which means substituting it for y in r, thus obtaining the contractum $\{\mathtt{D}\langle\{u/x\}t\rangle/y\}r$. Both the distant context and the term $\{u/x\}t$ may be duplicated, or, on the contrary, simply erased, such as in this example:

$$t_0 = (\lambda x.x(\mathtt{I}, y.y))(\mathtt{I}, z'.z) \to_{\mathtt{dn}} \{\{\mathtt{I}/x\}(x(\mathtt{I}, y.y))/z'\}z = \{\mathtt{I}(\mathtt{I}, y.y)/z'\}z = z$$

Although the original CBN calculus ΛJ_n is based on rules β and π, the distant CBN calculus λJ_n integrates a different permutative rule $\mathtt{p2}$ to β, in the sense that $\mathtt{d}\beta$ can be generated by several $\mathtt{p2}$-steps followed by a β-step, where:

$$(\textbf{Permutative } \mathtt{p2}\textbf{-Rule}) \quad t(u, y.\lambda x.r) \mapsto_{\mathtt{p2}} \lambda x.t(u, y.r).$$

Integrating the π-rule to define a distant CBN calculus would give instead the alternative rule $\mathtt{D}\langle\lambda x.t\rangle(u, y.r) \mapsto \mathtt{D}\langle\{\{u/x\}t/y\}r\rangle$, where the distant context $\mathtt{D}$ is neither duplicated nor erased: there is exactly one occurrence of $\mathtt{D}$ in the contractum. But this feature corresponds to a CBV behavior, and is not quantitatively well-behaved for a CBN semantics (see [18]). This same remark is true for the original CBN calculus ΛJ_n.

▶ **Definition 2.** *The **CBV calculus** λJ_v is defined by the grammar $\mathtt{T}_J$ equipped with the **distant call-by-value** reduction relation $\to_{\mathtt{dv}}$. The relation $\to_{\mathtt{dv}}$ is generated by the closure under all contexts $\mathtt{C}$ of the reduction rule: $\mathtt{D}\langle\lambda x.t\rangle(u, y.r) \mapsto_{\mathtt{d}\beta_{\mathtt{v}}} \mathtt{D}\langle\{\{u\backslash\!\backslash x\}t\backslash\!\backslash y\}r\rangle$.*

Notice that we also use a distance semantics for CBV, whereas the original ΛJ_v-calculus relies on rules π and $\beta_{\mathtt{v}}$ ($\mathtt{d}\beta_{\mathtt{v}}$ where $\mathtt{D}$ is empty). Both reductions $\to_{\mathtt{dn}}$ and $\to_{\mathtt{dv}}$ enjoy confluence. It is also worth noticing that CBN and CBV reducible expressions (*a.k.a. redexes*) are the same. In particular, no $\mathtt{d}\beta_{\mathtt{v}}$-redex is stuck because the argument is not a value. This is a major difference with most CBV calculi. However, the right-hand sides of the rules $\mathtt{d}\beta$ and $\mathtt{d}\beta_{\mathtt{v}}$ differ in two ways: the kind of substitution used (right for CBN, left for CBV) and the fact that $\mathtt{D}$ might be erased or duplicated in CBN, but not in CBV. To illustrate the first difference, we sketch the CBV reduction of the previous term t_0:

$$\begin{aligned} t_0 &= (\lambda x.x(\mathtt{I}, y.y))(\mathtt{I}, z'.z) \\ &\mapsto_{\mathtt{d}\beta_{\mathtt{v}}} \{\{\mathtt{I}\backslash\!\backslash x\}(x(\mathtt{I}, y.y))\backslash\!\backslash z'\}z = \{\mathtt{I}(\mathtt{I}, y.y)\backslash\!\backslash z'\}z = \mathtt{I}(\mathtt{I}, y.z) = (\lambda x.x)(\mathtt{I}, y.z) \\ &\mapsto_{\mathtt{d}\beta_{\mathtt{v}}} \{\{\mathtt{I}\backslash\!\backslash x\}x\backslash\!\backslash y\}z = \{\mathtt{I}\backslash\!\backslash y\}z = z \end{aligned}$$

In line two, the substitutions are exactly the ones already detailed after the definition of left substitution. The two (nested) left substitutions on the last line act on values, so that they fall back to the usual right substitution. While in CBN a single step was sufficient to reach the normal form z, in CBV we need one additional step.

As usual, CBV does not duplicate computations (outside abstractions), but tries to reduce every argument to a value, and this may create divergent computations. Take for instance $t_1 = \delta(\delta, z.y)$. In CBN, t_1 normalizes to y, while in CBV t_1 loops indefinitely.

$$\begin{aligned} &(\text{CBN}) \quad \delta(\delta, z.y) \mapsto_{\mathtt{d}\beta} \{\{\delta/x\}(x(x, z.z))/z\}y = \{\delta(\delta, z.z)/z\}y = y \\ &(\text{CBV}) \quad \delta(\delta, z.y) \mapsto_{\mathtt{d}\beta_{\mathtt{v}}} \{\{\delta\backslash\!\backslash x\}(x(x, z.z))\backslash\!\backslash z\}y \\ &\qquad = \{\{\delta/x\}(x(x, z.z))\backslash\!\backslash z\}y = \{\delta(\delta, z.z)\backslash\!\backslash z\}y = \delta(\delta, z.\{z\backslash\!\backslash z\}y) = \delta(\delta, z.y) \end{aligned}$$

We illustrate the different uses of distant contexts in CBN and CBV on a last example. Let $t_2 = x_1(u, x_2.\lambda x.x)(v, y.y(y, z.z))$, that can be written as $\mathtt{D}\langle\lambda x.x\rangle(v, y.y(y, z.z))$, where $\mathtt{D} = x_1(u, x_2.\Diamond)$. Using the distant semantics, the term v turns out to be an argument of the function $\lambda x.x$, and thus the redex can be fired, although both terms are separated by the (distant) context $\mathtt{D}$:

$$\begin{array}{lll}
\text{(CBN)} & t_2 \to^*_{\mathtt{d}\beta} & \{\mathtt{D}\langle\{v/x\}x\rangle/y\}(y(y,z.z)) = \{\mathtt{D}\langle v\rangle/y\}(y(y,z.z)) \\
 & = & \mathtt{D}\langle v\rangle(\mathtt{D}\langle v\rangle, z.z) = x_1(u,x_2.v)(x_1(u,x_2.v),z.z) \\
\text{(CBV)} & t_2 \to^*_{\mathtt{d}\beta_{\mathtt{v}}} & \mathtt{D}\langle\{\{v\backslash\!\!\backslash x\}x\backslash\!\!\backslash y\}(y(y,z.z))\rangle = \mathtt{D}\langle\{v\backslash\!\!\backslash y\}(y(y,z.z))\rangle \\
 & = & \mathtt{D}\langle v(v,z.z)\rangle = x_1(u,x_2.v(v,z.z))
\end{array}$$

The way distant contexts are handled in both variants is reminiscent of how the result of the β-reduction is treated by the two different substitution operations, as the applications inside the distant context are duplicated or erased only in CBN. This difference is justified by the use of distinct underlying permutation rules: p2 for CBN and π for CBV.

As a last remark, notice that $\beta_{\mathtt{v}}$-reduction preserves π-normal forms. Thus, an alternative to distant CBV reduction is to "pre-process" all terms by using full π-normalization, so that neither distance nor permutations are needed for the CBV computation, which is then defined only on terms in π-nf. Yet, instead of computing only with π-nf, we choose to work in a more general framework with arbitrary terms, by applying only those permutations that are necessary to fire β-redexes, following the same philosophy used in CBN.

3 Solvability for Generalized Applications

In this section we give definitions of solvability for CBN and CBV by using the key notion of *head context*. In contrast to the λ-calculus, the syntax of generalized applications makes the identification of the *head* of a term very subtle. In particular, whereas it is possible to use vectorial meta-notations in the λ-calculus, we must use inductive definitions in this framework. **Head contexts** are given by the following grammar:

$$\mathtt{H} ::= \Diamond \mid \lambda x.\mathtt{H} \mid \mathtt{H}(u, x.\mathtt{H}'\langle\!\langle x\rangle\!\rangle) \mid t(u, x.\mathtt{H})$$

While, in general, there are several possibilities to decompose a term into a head context surrounding a subterm, there is a closely related notion of **head variable**, which deterministically distinguishes a particular variable in each term:

$$\frac{}{\mathrm{hv}(x) = x} \qquad \frac{\mathrm{hv}(t) = x}{\mathrm{hv}(\lambda y.t) = x} \qquad \frac{\mathrm{hv}(r) = y \qquad \mathrm{hv}(t) = x}{\mathrm{hv}(t(u, y.r)) = x} \qquad \frac{\mathrm{hv}(r) = x \qquad x \neq y}{\mathrm{hv}(t(u, y.r)) = x}$$

In the third rule we assume w.l.o.g. that y is not bound in r. Notice that the head variable may be either free or bound, since y can be equal to x in the second rule. To understand the last two rules, we use the previous analogy with let-bindings. To an application $t(u, y.r)$ corresponds a binding `let` y = tu `in` r, and to find the head variable of this term, we look inside r. For instance, the head variable of `let` x = zz `in` y, corresponding to $z(z, x.y)$, is y. But if we take `let` x = zz `in` x, corresponding to $z(z, x.x)$, its head variable is z. Thus, the head variable of a term with generalized applications is the head variable of the corresponding term where all the let-binding have been unfolded. In the example, z is the head variable of $z(z, x.x)$ because x is itself the head variable of the subterm x inside the continuation.

Given a term t verifying $\mathrm{hv}(t) = x$, there is a unique head context $\mathtt{H}$ such that (1) $t = \mathtt{H}\langle x\rangle$, or (2) $t = \mathtt{H}\langle\!\langle x\rangle\!\rangle$ if $x \in \mathrm{fv}(t)$. Thus for example, given $t_3 := z(z, x.y)$, we have $\mathrm{hv}(t_3) = y$ as well as $t_3 = \mathtt{H}\langle\!\langle y\rangle\!\rangle$ with $\mathtt{H} = z(z, x.\Diamond)$. Given $t'_3 := z(z, x.x)$, we have $\mathrm{hv}(t'_3) = z$ as well as $t'_3 = \mathtt{H}'\langle\!\langle z\rangle\!\rangle$ with $\mathtt{H}' = \Diamond(z, x.\mathtt{H}_0\langle\!\langle x\rangle\!\rangle)$ and $\mathtt{H}_0 = \Diamond$. An example where the head variable is bound is $\mathrm{hv}(\lambda y.t_3) = y$, where $\lambda y.t_3 = \mathtt{H}''\langle y\rangle$ and $\mathtt{H}'' = \lambda y.z(z, x.\Diamond)$.

▶ **Definition 3** (Solvability). *Let $t \in \mathtt{T}_J$. Then:*
t ***is CBN-solvable*** *iff there is a head (resp. distant) context* $\mathtt{H}$ *and* $\mathtt{D}$ *such that* $\mathtt{H}\langle t\rangle \to^*_{\mathtt{dn}} \mathtt{D}\langle\mathtt{I}\rangle$.
t ***is CBV-solvable*** *iff there is a head context* $\mathtt{H}$ *such that* $\mathtt{H}\langle t\rangle \to^*_{\mathtt{dv}} \mathtt{I}$.

In the CBN λ-calculus, several equivalent definitions of solvability for a λ-term M coexist. In particular: (1) there is a head context $\mathtt{H}$ such that $\mathtt{H}\langle M\rangle \to^*_\beta \mathtt{I}$; (2) for all λ-term N, there is a head context $\mathtt{H}$ such that $\mathtt{H}\langle M\rangle \to^*_\beta N$. The head contexts here are the usual head contexts of the λ-calculus. The proof that (1) implies (2) is trivial, since $\mathtt{I}N \to_\beta N$ always hold in the CBN λ-calculus. On the contrary, in Plotkin's original CBV, the two formulations are not equivalent [20], since $\mathtt{I}N$ only reduces to N when N is a value. In our CBV framework, the equivalence of these alternative definitions is straightforward: for any $\mathtt{T}_J$-term u, $\mathtt{I}(u, z.z) \mapsto_{\mathtt{d}\beta_\mathtt{v}} u$ always holds. Moreover, our definition of CBN-solvability for a term t is equivalent to an alternative one stating that for all $\mathtt{T}_J$-term u there exist $\mathtt{H}$ and $\mathtt{D}$ such that $\mathtt{H}\langle t\rangle \to^*_{\mathtt{dn}} \mathtt{D}\langle u\rangle$. Indeed, $\mathtt{D}\langle\mathtt{I}\rangle(u, z.z) \mapsto_{\mathtt{d}\beta} \mathtt{D}\langle u\rangle$ for any u.

Notice in our definition of CBN-solvability that the reduction yields an identity plugged inside a distant context, and not just an identity alone. Take *e.g.* the term $t_4 = \Omega(y, z.\mathtt{I})$ containing a non-relevant continuation, as $z \notin \mathrm{fv}(\mathtt{I})$. In the λ-calculus, t_4 translates to $(\lambda z.\mathtt{I})(\Omega y)$, which is solvable since $(\lambda z.\mathtt{I})(\Omega y) \to_\beta \mathtt{I}$. This suggests introducing a garbage collection-like rule for generalized applications which reduces in this case $\Omega(y, z.\mathtt{I}) \to_{\mathtt{gc}} \mathtt{I}$. This would be consistent with different models of CBN, such as our quantitative type system. However, we prefer to avoid such ad-hoc solution, which can be simply seen as an implementation detail, as it does not change the operational and denotational behavior of terms.

Now, why does our notion of CBV solvability not use this distant context? Take again the term $t_4 = \Omega(y, z.\mathtt{I})$ and its translated λ-term $(\lambda z.\mathtt{I})(\Omega y)$. CBV reduction in the λ-calculus loops on the argument Ωy, that could only be erased if Ωy is reduced to a value. Therefore, having a definition of solvability which reduces to $\mathtt{D}\langle\mathtt{I}\rangle$ in CBV would be too liberal, and incoherent with the λ-calculus and its associated models.

Although the two definitions of CBN/CBV solvability are slightly different, they both share the same notion of head context, which is independent from the operational semantics.

Head reduction in the λ-calculus is the reduction relation generated by the closure of the rule β under *head* contexts. A key (operational) property is that a term t is CBN solvable *iff* t normalizes for the head reduction, that is why we say that it is a *solvable reduction*. In Subsec. 4.1, we define a notion of solvable reduction for λJ_n, and show that it also has this crucial property. For λJ_v, we will see in Subsec. 5.1 that the solvable relation is bigger than plain CBN solvable reduction. Nevertheless, the solvable reduction we give mimics the behavior of a solvable reduction for the CBV λ-calculus.

4 Call-by-Name Solvability

This section is organized in two parts. We first give an operational characterization of solvability through a solvable reduction relation, and then a quantitative type system providing a logical characterization of it.

4.1 Operational Characterization of CBN Solvability

The following solvable reduction relation for λJ_n plays the same role as the head reduction plays for the λ-calculus.

▶ **Definition 4.** *The **CBN solvable reduction** $\to_{\mathtt{sn}}$ is defined as the closure of the following reduction rule under head contexts.*

$$\mathtt{D}\langle \lambda x.t\rangle(u, y.\mathtt{H}\langle\!\langle y\rangle\!\rangle) \mapsto_{\mathtt{d}\beta\mathtt{h}} \{\mathtt{D}\langle\{u/x\}t\rangle/y\}\mathtt{H}\langle\!\langle y\rangle\!\rangle$$

Notice that the solvable reduction is not based on the full $\mathtt{d}\beta$ rule, as it only reduces redexes to which (one of the hereditary) head variables is bound. Thus, the term $t_4 = \Omega(y, z.\mathtt{I})$ is $\mathtt{sn}$-normal but not $\mathtt{d}\beta$-normal.

▶ **Lemma 5.** *The following grammar* $\mathrm{NF}_{\mathtt{sn}}$ *characterizes* $\mathtt{sn}$*-nfs.*

$$\begin{array}{rrl} \textit{(CBN Neutral Normal Contexts)} & \mathtt{G} ::= & \Diamond \mid \mathtt{G}(u, x.\mathtt{G}\langle\!\langle x\rangle\!\rangle) \mid t(u, y.\mathtt{G}) \\ \textit{(CBN Solvable Normal Terms)} & \mathrm{NF}_{\mathtt{sn}} ::= & x \mid \lambda x.\,\mathrm{NF}_{\mathtt{sn}} \mid \mathtt{G}\langle\!\langle x\rangle\!\rangle(u, y.\,\mathrm{NF}_{\mathtt{sn}}) \\ & & \mid t(u, x.\,\mathrm{NF}_{\mathtt{sn}}) \;\; \textit{where } x \neq \mathrm{hv}(\mathrm{NF}_{\mathtt{sn}}) \end{array}$$

The previous grammar $\mathrm{NF}_{\mathtt{sn}}$ is used to show the following key result:

▶ **Lemma 6.** *Let t be an* $\mathtt{sn}$*-normalizable term. Then t is CBN solvable.*

To give some intuition about the proof, we know t $\mathtt{sn}$-normalizable implies there is some $\mathtt{sn}$-normal form t' such that $t \to^*_{\mathtt{sn}} t'$. The goal of the proof is to construct a head context $\mathtt{H}$ such that $\mathtt{H}\langle t'\rangle \to^*_{\mathtt{dn}} \mathtt{D}\langle\mathtt{I}\rangle$ for some $\mathtt{D}$. This is done by induction on the grammar $\mathrm{NF}_{\mathtt{sn}}$. Therefore t is solvable since $\mathtt{H}\langle t\rangle \to^*_{\mathtt{dn}} \mathtt{H}\langle t'\rangle \to^*_{\mathtt{dn}} \mathtt{D}\langle\mathtt{I}\rangle$.

▶ **Example 7.** Let $t_5 = y_1(\mathtt{I}, z_1.x)(y_2(\mathtt{I}, z_2.z_2), z_3.\lambda y.z_3) \in \mathrm{NF}_{\mathtt{sn}}$. Notice that $\mathrm{hv}(t_5) = x$. We take $\mathtt{H} = (\lambda x.\Diamond)(\mathbf{o}^1, z.z)(\mathtt{I}, z.z)$ (remember that $\mathbf{o}^1 = \lambda x_1 \lambda x_0.x_0$). Then,

$$\begin{aligned} \mathtt{H}\langle t_5\rangle &= (\lambda x.y_1(\mathtt{I}, z_1.x)(y_2(\mathtt{I}, z_2.z_2), z_3.\lambda y.z_3))(\mathbf{o}^1, z.z)(\mathtt{I}, z.z) \\ &\to_{\mathtt{dn}} y_1(\mathtt{I}, z_1.\,\mathbf{o}^1)(y_2(\mathtt{I}, z_2.z_2), z_3.\lambda y.z_3)(\mathtt{I}, z.z) \\ &\to_{\mathtt{dn}} (\lambda y.y_1(\mathtt{I}, z_1.\,\mathbf{o}^0))(\mathtt{I}, z.z) \\ &\to_{\mathtt{dn}} y_1(\mathtt{I}, z_1.\,\mathbf{o}^0) \end{aligned}$$

Taking $\mathtt{D} = y_1(\mathtt{I}, z_1.\Diamond)$, we get a term of the expected form $\mathtt{D}\langle\mathtt{I}\rangle$ (since $\mathbf{o}^0 = \mathtt{I}$).

4.2 Logical Characterization of CBN Solvability

We now give a type system, called $\cap N$, in which typability and normalization of solvable reduction coincide, *i.e.* not only does typability imply normalization, but the converse implication also holds. In system $\cap N$, terms can be assigned several types. However, they are not typed with sets – which would give an idempotent intersection type system [13] – , but with multisets, which give a *quantitative* flavor to the type system. Also, non-idempotent types justify the use of rule $\mathtt{p2}$, instead of π, to obtain a distant notion for CBN with generalized applications which is quantitatively well behaved ([18]). Moreover, as in other calculi with non-idempotent intersection types, the proofs of normalization are notably simplified: reducibility or computability arguments can be replaced by simple combinatorial proofs based on the fact that the size of the type derivations strictly decreases along each solvable $\mathtt{sn}$-step.

Given a countable infinite set BTV of base type variables $a, b, c, \ldots$, we define the following sets of types:

$$\begin{array}{rrcl} \textbf{(Types)} & \sigma, \tau, \rho & ::= & a \in BTV \mid \mathcal{M} \to \sigma \\ \textbf{(Multiset types)} & \mathcal{M}, \mathcal{N} & ::= & [\sigma_i]_{i \in I} \text{ where } I \text{ is a finite set} \end{array}$$

The empty multiset is denoted $[\,]$. **Environments**, written Γ, Δ, Λ, are functions from variables to multiset types assigning the empty multiset to all but a finite set of variables. A **typing** is a pair of an environment and a type. The domain of Γ is given by $\mathrm{dom}(\Gamma) := \{x \mid \Gamma(x) \neq [\,]\}$. The **union of environments**, written $\Gamma \wedge \Delta$, is defined by $(\Gamma \wedge \Delta)(x) := \Gamma(x) \sqcup \Delta(x)$, where $\sqcup$ denotes multiset union. This notion is extended to several environments as expected, so that $\wedge_{i \in I} \Gamma_i$ denotes a finite union of environments ($\wedge_{i \in I} \Gamma_i$ is to be understood as the empty environment when $I = \emptyset$). We write $\Gamma \setminus\!\!\setminus x$ for the environment such that $(\Gamma \setminus\!\!\setminus x)(y) = \Gamma(y)$ if $y \neq x$ and $(\Gamma \setminus\!\!\setminus x)(x) = [\,]$. We write $\Gamma; \Delta$ for $\Gamma \wedge \Delta$ when $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Delta) = \emptyset$. A **sequent** has the form $\Gamma \vdash t : \sigma$, where (Γ, σ) is a typing and t is a term.

$$\frac{}{x : [\sigma] \vdash x : \sigma}\ (\textsc{var}) \qquad \frac{\Gamma; x : \mathcal{M} \vdash t : \sigma}{\Gamma \vdash \lambda x.t : \mathcal{M} \to \sigma}\ (\textsc{abs}) \qquad \frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{\wedge_{i \in I} \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (\textsc{many})$$

$$\frac{\Gamma \vdash t : [\mathcal{M}_i \to \sigma_i]_{i \in I} \qquad \Delta \vdash u : \sqcup_{i \in I} \mathcal{M}_i \qquad \Lambda; x : [\sigma_i]_{i \in I} \vdash r : \tau}{\Gamma \wedge \Delta \wedge \Lambda \vdash t(u, x.r) : \tau}\ (\textsc{app})$$

Figure 1 System $\cap N$.

The quantitative type system $\cap N$ is defined in Fig. 1. Notice that the system is relevant, as there is no weakening. It is a natural extension of Gardner's [21] and De Carvalho's [14] systems to generalized applications. Rule (MANY) may give the empty multiset to any term (case $I = \emptyset$), so being typable with $[\,]$ means in fact being untyped. The interesting rule is (APP), where both t and u are assigned multiset types, since x is not necessarily linear in r. Because u is the argument of t, it is assigned all the types on the left of the arrow of t. For $n \in \mathbb{N}$, we write $\Gamma \Vdash^n_{\cap N} t : \sigma$ or simply $\Gamma \Vdash^n t : \sigma$ if there is a derivation in system $\cap N$ ending in $\Gamma \vdash t : \sigma$ and containing n occurrences of rule (APP). This derivation measure is sufficient to capture the fact that each $\mathtt{sn}$-step deletes at least one (APP) rule (*c.f.* Lem. 10).

▶ **Example 8.** Take again $t_4 = \Omega(y, z.\mathtt{I})$ (we expand $\mathtt{I}$ to $\lambda x.x$ in the derivation). Although the evaluation of the subterm Ω is not terminating (and thus Ω can only be typed with the empty multiset), t_4 is typable:

$$\frac{\dfrac{}{\vdash \Omega : [\,]}\ (\textsc{many}) \qquad \dfrac{}{\vdash y : [\,]}\ (\textsc{many}) \qquad \dfrac{\dfrac{}{x : [\sigma] \vdash x : \sigma}\ (\textsc{var})}{\vdash \lambda x.x : [\sigma] \to \sigma}\ (\textsc{abs})}{\vdash \Omega(y, z.\lambda x.x) : [\sigma] \to \sigma}\ (\textsc{app})$$

We now prove that terms typable in $\cap N$ are exactly the ones that normalize with $\to_{\mathtt{sn}}$. The proof relies on two key lemmas.

1. *Weighted subject reduction* does not only state that the typing of a term is preserved along reduction – as usual – , but also that the size of the typing derivation decreases at each $\mathtt{sn}$-step. From this we can deduce that typable terms normalize with $\to_{\mathtt{sn}}$.
2. *Subject expansion* is the opposite of subject reduction: if t reduces to a term t' and t' is typable, then t is also typable with the same typing as t'. It is also possible to attach quantitative information to the subject expansion lemma, proving that the size of the type derivation of t' is smaller than the one of t, but this will not be useful here.

Subject reduction and subject expansion give soundness and completeness of $\to_{\mathtt{dn}}$ w.r.t. to the type system. This is a peculiarity of intersection type systems, which makes it possible to derive a denotational semantics from the typing itself. In the case of quantitative type systems, it is a relational model, *i.e.* a model in the category **Rel** of sets and relations [9].

Soundness

To prove subject reduction, we first need to prove a *substitution lemma*, as usual. Because we are in a quantitative model, this lemma also relates the sizes of the corresponding derivations.

▶ **Lemma 9** (Substitution for $\cap N$). *If $\Gamma; x : \mathcal{M} \Vdash^{n}_{\cap N} t : \sigma$ and $\Delta \Vdash^{m}_{\cap N} u : \mathcal{M}$, then there is a derivation $\Gamma \wedge \Delta \Vdash^{m+n}_{\cap N} \{u/x\}t : \sigma$.*

▶ **Lemma 10** (Weighted Subject Reduction for $\cap N$). *If $\Gamma \Vdash^{n_1}_{\cap N} t_1 : \sigma$ and $t_1 \to_{\mathtt{sn}} t_2$, then $\Gamma \Vdash^{n_2}_{\cap N} t_2 : \sigma$ with $n_1 > n_2$.*

Proof. The proof is by induction on the reduction step $t_1 \to_{\mathtt{sn}} t_2$. The base case is $t_1 = \mathtt{D}\langle \lambda x.t\rangle(u, y.\mathtt{H}\langle\langle y\rangle\rangle) \mapsto_{\mathtt{sn}} \{\mathtt{D}\langle\{u/x\}t\rangle/y\}\mathtt{H}\langle\langle y\rangle\rangle = t_2$, which we decompose into a series of p2-permutation steps followed by a $\beta\mathtt{h}$-step (a $\mathtt{d}\beta\mathtt{h}$-step with an empty distant context): $t_1 \mapsto^{*}_{\mathtt{p2}} (\lambda x.\mathtt{D}\langle t\rangle)(u, y.\mathtt{H}\langle\langle y\rangle\rangle) \mapsto_{\beta\mathtt{h}} t_2$. We prove subject reduction independently for p2 and βh. For the last one, we apply the substitution lemma twice. The permutative and the inductive cases (reduction under head contexts) are straightforward by the induction hypothesis. ◀

The size of type derivations is a natural number decreasing at every step, so that soundness is a direct corollary. Notice that no reducibility proof is needed.

▶ **Corollary 11** (Soundness for λJ_n). *If $\Gamma \Vdash^{n}_{\cap N} t : \sigma$, then t is* sn*-normalizable and the number of* sn*-steps needed to normalize t is bounded by n.*

Completeness

To show that every sn-normalizable term is typable, we need subject expansion. This property, like subject reduction, needs a preliminary lemma, this time *anti-substitution.*

▶ **Lemma 12** (Anti-Substitution for $\cap N$). *If $\Gamma \Vdash \{u/x\}t : \sigma$, then there exists Γ_t, Γ_u and $\mathcal{M}$ such that $\Gamma_t; x : \mathcal{M} \Vdash t : \sigma$, $\Gamma_u \Vdash u : \mathcal{M}$ and $\Gamma = \Gamma_t \wedge \Gamma_u$.*

▶ **Lemma 13** (Subject Expansion for $\cap N$). *If $\Gamma \Vdash_{\cap N} t_2 : \sigma$ and $t_1 \to_{\mathtt{dn}} t_2$, then $\Gamma \Vdash_{\cap N} t_1 : \sigma$.*

Proof. Notice that the statement is about full dn reduction, which is useful in the proof of Thm. 16. The proof is by induction on $t_1 \to_{\mathtt{dn}} t_2$ and uses the anti-substitution lemma. ◀

Another component of the completeness proof is the fact that sn-normal forms are typable.

▶ **Lemma 14** (Typing sn-nfs). *Let $t \in \mathrm{NF}_{\mathtt{sn}}$. Then there exists σ such that*
1. *If $t = \mathtt{H}\langle\langle x\rangle\rangle$ for some x, then there is τ such that $x : [\tau] \Vdash t : \sigma$.*
2. *Otherwise, $\emptyset \Vdash t : \sigma$.*

▶ **Corollary 15** (Completeness for λJ_n). *Let $t \in \mathtt{T}_J$ be* sn*-normalizable. Then t is typable in system $\cap N$.*

Proof. By definition, the term t is reducible to a sn-normal form t'. By Lem. 14, t' is typable. Subject Expansion gives typability of t. ◀

Characterization of CBN Solvability

We can now derive the main theorem of this section.

▶ **Theorem 16** (CBN Characterization). *Let $t \in \mathtt{T}_J$. Then t is CBN solvable iff t is $\cap N$-typable iff t is $\mathtt{sn}$-normalizable.*

Proof. Normalizable $\implies$ solvable holds by Lem. 6. Typable $\implies$ normalizable holds by Cor. 11. For solvable $\implies$ typable: take t solvable, so that there are contexts $\mathtt{H}, \mathtt{D}$ such that $\mathtt{H}\langle t\rangle \rightarrow^*_{\mathtt{dn}} \mathtt{D}\langle \mathtt{I}\rangle$. Since $\mathtt{D}\langle \mathtt{I}\rangle$ is $\cap N$-typable by Lem. 14, and the system $\cap N$ satisfies subject expansion (Lem. 13), then $\mathtt{H}\langle t\rangle$ is $\cap N$-typable, which implies t is $\cap N$-typable. ◀

5 Call-by-Value Solvability

As in the previous section, we first give an operational characterization of solvability and then a quantitative type system characterizing it.

5.1 Operational Characterization of CBV Solvability

In CBN, the method to get the identity from a term plugged into a head context is to successively erase all the arguments, by replacing the head variable by a projection term $\mathbf{o}^n = \lambda x_n \ldots x_0.x_0$. But in CBV, arguments which are not values cannot be erased: in order to be erased they need to be *potentially valuable*.

▶ **Definition 17.** *A term t is **potentially valuable** iff there exist a distant context $\mathtt{D}$ and a value v such that $\mathtt{D}\langle t\rangle \rightarrow^*_{\mathtt{dv}} v$.*

The distant context in the previous definition can be seen as a list of substitutions – eventually affecting the free variables of t – used to transform t into a value.

▶ **Example 18.** Let $t_6 = x(\Omega, z.z)$. In CBN, it is sufficient to take the head context $\mathtt{H} = \mathtt{I}(\mathbf{o}^1, x.\Diamond)$ so that $\mathtt{H}\langle t_6\rangle \rightarrow_{\mathtt{dn}} \mathbf{o}^1(\Omega, z.z) \rightarrow_{\mathtt{dn}} \mathtt{I}$ simply erases the diverging term Ω. In CBV, however, this is not possible since $\mathtt{H}\langle t_6\rangle \rightarrow_{\mathtt{dv}} \mathbf{o}^1(\Omega, z.z) \rightarrow_{\mathtt{dv}} \delta(\delta, x.\mathtt{I})$, which diverges. The term t_6 is only solvable in CBN. On the contrary, the term $x(\lambda y.\Omega, z.z)$ is solvable in both CBN and CBV because the argument $\lambda y.\Omega$ can be erased.

Interestingly, there is a (non-deterministic) reduction relation $\rightarrow_{\mathtt{pv}}$ such that the normalizing terms for $\rightarrow_{\mathtt{pv}}$ are exactly the potentially valuable terms (*c.f.* Thm. 34). It is in fact a *weak* reduction relation in which reduction can occur anywhere but below abstractions. We detail this result before tackling the CBV solvable reduction.

▶ **Definition 19.** *The **valuable reduction relation** $\rightarrow_{\mathtt{pv}}$ is defined by the following rules:*

$$\frac{t \mapsto_{\mathtt{d}\beta_{\mathtt{v}}} t'}{t \rightarrow_{\mathtt{pv}} t'} \qquad \frac{t \rightarrow_{\mathtt{pv}} t'}{t(u, y.r) \rightarrow_{\mathtt{pv}} t'(u, y.r)} \qquad \frac{u \rightarrow_{\mathtt{pv}} u'}{t(u, y.r) \rightarrow_{\mathtt{pv}} t(u', y.r)} \qquad \frac{r \rightarrow_{\mathtt{pv}} r'}{t(u, y.r) \rightarrow_{\mathtt{pv}} t(u, y.r')}$$

▶ **Lemma 20.** *Consider the following grammar:*

$$\begin{array}{llcl} \textit{(Valuable Neutral Normal Terms)} & \mathrm{NE}_{\mathtt{pv}} & ::= & x \mid \mathrm{NE}_{\mathtt{pv}}(\mathrm{NF}_{\mathtt{pv}}, y.\,\mathrm{NE}_{\mathtt{pv}}) \\ \textit{(Valuable Normal Terms)} & \mathrm{NF}_{\mathtt{pv}} & ::= & x \mid \mathrm{NE}_{\mathtt{pv}}(\mathrm{NF}_{\mathtt{pv}}, y.\,\mathrm{NF}_{\mathtt{pv}}) \mid \lambda x.t \end{array}$$

Then, $t \in \mathrm{NF}_{\mathtt{pv}}$ iff t is in $\mathtt{pv}$-normal form.

This grammar is used to show the following property, whose converse is obtained in Thm. 34.

▶ **Lemma 21.** *Let t be a* $\texttt{pv}$*-normalizable term. Then t is potentially valuable.*

Note that being potentially valuable is weaker than being solvable, because values are always potentially valuable but not necessarily solvable, like $\lambda x.\Omega$. We are now ready to build the solvable reduction on top of the valuable one.

▶ **Definition 22.** *The **CBV solvable reduction relation** $\to_{\texttt{sv}}$ is defined as follows:*

$$\frac{t \mapsto_{\texttt{d}\beta_{\texttt{v}}} t'}{t \to_{\texttt{sv}} t'} \qquad \frac{t \to_{\texttt{sv}} t'}{\lambda x.t \to_{\texttt{sv}} \lambda x.t'}$$

$$\frac{t \to_{\texttt{pv}} t'}{t(u, x.r) \to_{\texttt{sv}} t'(u, x.r)} \qquad \frac{u \to_{\texttt{pv}} u'}{t(u, x.r) \to_{\texttt{sv}} t(u', x.r)} \qquad \frac{r \to_{\texttt{sv}} r'}{t(u, x.r) \to_{\texttt{sv}} t(u, x.r')}$$

An equivalent formulation can be given by the closure of $\texttt{d}\beta_{\texttt{v}}$ under head contexts, plus the first and second rules for closure under application. With these rules, we make sure that in an application $t(u, x.r)$, the subterms t and u are $\texttt{pv}$-normalizable, and thus potentially valuable. In case there is a divergent term in u or t, the solvable reduction will diverge. This relation is strictly bigger than the CBN solvable relation $\to_{\texttt{sn}}$, as it diverges on more terms.

▶ **Lemma 23.** *Let us consider the following grammar:*

$$\textbf{\textit{(CBV Solvable Normal Terms)}} \qquad \mathrm{NF}_{\texttt{sv}} ::= x \mid \lambda x.\, \mathrm{NF}_{\texttt{sv}} \mid \mathrm{NE}_{\texttt{pv}}(\mathrm{NF}_{\texttt{pv}}, y.\, \mathrm{NF}_{\texttt{sv}})$$

Then, $t \in \mathrm{NF}_{\texttt{sv}}$ iff t is in $\texttt{sv}$*-normal form. Notice that $\mathrm{NF}_{\texttt{sv}} \subset \mathrm{NF}_{\texttt{pv}}$.*

▶ **Lemma 24.** *Let t be an* $\texttt{sv}$*-normalizable term. Then t is CBV solvable.*

A hint of the proof: since t is $\texttt{sv}$-normalizable, there is some $\texttt{sv}$-normal form t such that $t \to^*_{\texttt{sv}} t'$. We must construct a head context $\texttt{H}$ such that $\texttt{H}\langle t' \rangle \to^*_{\texttt{dv}} \texttt{I}$. This is done by induction on the grammar $\mathrm{NF}_{\texttt{sv}}$. Therefore, t is solvable since $\texttt{H}\langle t \rangle \to^*_{\texttt{dv}} \texttt{H}\langle t' \rangle \to^*_{\texttt{dv}} \texttt{I}$. A difference with Lem. 6 for CBN is that the head context must erase all applications of t', even the ones which are non-relevant, making the proof more involved.

▶ **Example 25.** Once again, the goal is to construct a head context for the $\texttt{sv}$-normal form of t. Take for instance the term $t_5 = y_1(\texttt{I}, z_1.x)(y_2(\texttt{I}, z_2.z_2), z_3.\lambda y.z_3)$ from Ex. 7, also in $\mathrm{NF}_{\texttt{sv}}$. This time, we take $\texttt{H} = (\lambda y_1.\lambda x.\lambda y_2.\Diamond)(\mathbf{o}^1, z.z)(\mathbf{o}^1, z.z)(\mathbf{o}^1, z.z)(\texttt{I}, z.z)$. Indeed,

$$\begin{aligned}
\texttt{H}\langle t_5 \rangle &= (\lambda y_1.\lambda x.\lambda y_2.y_1(\texttt{I}, z_1.x)(y_2(\texttt{I}, z_2.z_2), z_3.\lambda y.z_3))(\mathbf{o}^1, z.z)(\mathbf{o}^1, z.z)(\mathbf{o}^1, z.z)(\texttt{I}, z.z) \\
&\to^3_{\texttt{dv}} \mathbf{o}^1(\texttt{I}, z_1.\, \mathbf{o}^1)(\mathbf{o}^1(\texttt{I}, z_2.z_2), z_3.\lambda y.z_3)(\texttt{I}, z.z) \to^3_{\texttt{dv}} \mathbf{o}^1(\mathbf{o}^1(\texttt{I}, z_2.z_2), z_3.\lambda y.z_3)(\texttt{I}, z.z) \\
&\to_{\texttt{dv}} \mathbf{o}^1(\texttt{I}, z_3.\lambda y.z_3)(\texttt{I}, z.z) \to_{\texttt{dv}} (\lambda y.\, \mathbf{o}^0)(\texttt{I}, z.z) \to_{\texttt{dv}} \mathbf{o}^0 = \texttt{I}
\end{aligned}$$

5.2 Logical Characterization of CBV Solvability

We will now define a quantitative type system characterizing CBV solvability. The grammar of types is different from Subsec. 4.2, as multiset types are considered as types and in particular may also occur on the right hand-side of an arrow.

$$\begin{array}{llll}
\textbf{(Types)} & \sigma, \tau & ::= & a \in BTV \mid \mathcal{M} \mid \mathcal{M} \to \sigma \\
\textbf{(Multiset types)} & \mathcal{M}, \mathcal{N} & ::= & [\sigma_i]_{i \in I} \text{ where } I \text{ is a finite set}
\end{array}$$

$$\frac{}{x:\mathcal{M}\vdash x:\mathcal{M}}\ (\text{VAR}) \qquad \frac{(\Gamma_i; x:\mathcal{M}_i \vdash t:\sigma_i)_{i\in I}}{\wedge_{i\in I}\Gamma_i \vdash \lambda x.t : [\mathcal{M}_i \to \sigma_i]_{i\in I}}\ (\text{ABS})$$

$$\frac{\Gamma\vdash t:[\mathcal{M}\to\mathcal{N}] \qquad \Delta\vdash u:\mathcal{M} \qquad \Lambda; x:\mathcal{N}\vdash r:\sigma}{\Gamma\wedge\Delta\wedge\Lambda\vdash t(u,x.r):\sigma}\ (\text{APP})$$

Figure 2 System $\cap V$.

We use a unique type system $\cap V$, defined in Fig. 2, to characterize both potential valuability and solvability. The type system is inspired from [10]. Again, we write $\Gamma \Vdash^{n}_{\cap V} t:\sigma$ if the sequent $\Gamma\vdash t:\sigma$ is derivable in this system with a derivation of size n (containing n occurrences of (APP)). We will show that typability in $\cap V$ is equivalent to normalizability of the valuable reduction. To logically characterize solvable terms, we constraint typability to a particular set of types, where the empty multiset type cannot appear anymore on the right-hand sides of arrows. We take this idea from [5], where these types are called *solvable*.

▶ **Definition 26** (Solvable types). *A* ***solvable type*** *is not an empty multiset, and has no empty multiset on the right of an arrow. Formally,*

$$\begin{array}{lrcl} (\textbf{\textit{Solvable types}}) & \sigma^{\mathrm{s}},\tau^{\mathrm{s}} & ::= & a\in BTV \mid \mathcal{M}^{\mathrm{s}} \mid \mathcal{M}\to\sigma^{\mathrm{s}} \\ (\textbf{\textit{Solvable multiset types}}) & \mathcal{M}^{\mathrm{s}},\mathcal{N}^{\mathrm{s}} & ::= & [\sigma^{\mathrm{s}}_i]_{i\in I} \quad \textit{where } I \textit{ is a non-empty finite set} \end{array}$$

Unlike CBN, where the empty multiset $[\,]$ is used to mark *untyped* subterms, being typable in CBV with $[\,]$ is equivalent to being potentially valuable. The unsolvable term $\lambda x.\Omega$, for instance, can be typed with $[\,]$ by rule (ABS) with I empty. But it cannot be typed with any other type, and in particular with a solvable one, otherwise Ω would need to be typable. Notice also that the terms t and u in rule (APP) must always be typed, at least with type $[\,]$. That is why the term t_4 of Ex. 8, typable in $\cap N$, is not typable in $\cap V$.

We now prove that terms typable in $\cap V$ are exactly those that are normalizable for the valuable reduction, and among them, those typable with a solvable type are the ones normalizing for the solvable reduction. The proof method is the same as for CBN (Subsec. 4.2), but the statements cover both reduction relations at the same time.

Soundness

Soundness follows the same scheme used for CBN (no reducibility proof is needed): a weighted subject reduction property, based on a substitution lemma, is used to show that typability implies normalization.

▶ **Lemma 27** (Substitution for $\cap V$). *Let* $\Gamma; x:\mathcal{M} \Vdash^{n}_{\cap V} t:\sigma$ *and* $\Delta \Vdash^{m}_{\cap V} u:\mathcal{M}$.
- *If u is a value, then* $\Gamma\wedge\Delta \Vdash^{n+m}_{\cap V} \{u/x\}t:\sigma$.
- *For any u,* $\Gamma\wedge\Delta \Vdash^{n+m}_{\cap V} \{u\backslash\!\!\backslash x\}t:\sigma$.

▶ **Lemma 28** (Weighted Subject Reduction for $\cap V$). *Let* $\Gamma \Vdash^{n_1}_{\cap V} t_1:\sigma$ *and* $t_1 \to_{\mathtt{dv}} t_2$. *Then* $\Gamma \Vdash^{n_2}_{\cap V} t_2:\sigma$ *with* $n_1\geq n_2$. *Moreover:*
1. *If* $t_1 \to_{\mathtt{pv}} t_2$, *then* $n_1 > n_2$.
2. *If* $t_1 \to_{\mathtt{sv}} t_2$ *and σ is a solvable type, then* $n_1 > n_2$.

▶ **Corollary 29** (Soundness for $\cap V$). *Let $\Gamma \Vdash^{n}_{\cap V} t : \sigma$. Then,*

1. *The term t is* `pv`*-normalizing and the number of* `pv`*-steps needed to normalize t is bound by n.*
2. *If σ is a solvable type, then t is* `sv`*-normalizing and the number of* `sv`*-steps needed to normalize t is bound by n.*

Completeness

Completeness also follows the same scheme used for CBN: we show that normal forms are typable, together with a subject expansion property, based on an anti-substitution lemma.

▶ **Lemma 30** (Anti-substitution for $\cap V$).

- *If $\Gamma \Vdash_{\cap V} \{v/x\}t : \sigma$, then there are Γ_t, Γ_v and $\mathcal{M}$ such that $\Gamma_t; x : \mathcal{M} \Vdash_{\cap V} t : \sigma$, $\Gamma_v \Vdash_{\cap V} v : \mathcal{M}$ and $\Gamma = \Gamma_t \wedge \Gamma_v$.*
- *If $\Gamma \Vdash_{\cap V} \{u \backslash\backslash x\}t : \sigma$, then there are Γ_t, Γ_u and $\mathcal{M}$ such that $\Gamma_t; x : \mathcal{M} \Vdash_{\cap V} t : \sigma$, $\Gamma_u \Vdash_{\cap V} u : \mathcal{M}$ and $\Gamma = \Gamma_t \wedge \Gamma_u$.*

▶ **Lemma 31** (Subject Expansion for $\cap V$). *Let $\Gamma \Vdash_{\cap V} t_2 : \sigma$ and $t_1 \rightarrow_{\mathtt{dv}} t_2$. Then $\Gamma \Vdash_{\cap V} t_1 : \sigma$.*

▶ **Lemma 32** (Typing $\mathrm{NF}_{\mathtt{pv}}$-nfs). *Let $t \in \mathtt{T}_J$.*

1. *If $t \in \mathrm{NF}_{\mathtt{pv}}$, then there exists Γ such that $\Gamma \Vdash_{\cap V} t : [\,]$.*
2. *If $t \in \mathrm{NF}_{\mathtt{sv}}$, then there exist Γ and σ^{s} solvable such that $\Gamma \Vdash_{\cap V} t : \sigma^{\mathrm{s}}$.*

▶ **Corollary 33** (Completeness for $\cap V$). *Let $t \in \mathtt{T}_J$.*

1. *If t is* `pv`*-normalizing, then t is typable in $\cap V$.*
2. *If t is* `sv`*-normalizing, then t is typable in $\cap V$ with a solvable type.*

Characterization of CBV Solvability

We can now derive the main theorem of this section.

▶ **Theorem 34** (Characterization). *Let $t \in \mathtt{T}_J$. Then,*

- *t is potentially valuable iff t is $\cap V$-typable iff t is* `pv`*-normalizable, and*
- *t is CBV solvable iff t is $\cap V$-typable with a solvable type iff t is* `sv`*-normalizable.*

Proof. `pv`/`sv`-normalizable $\Longrightarrow$ potentially valuable/CBV solvable holds respectively by Lem. 21/Lem. 24. Typable/Typable with a solvable type $\Longrightarrow$ `pv`/`sv`-normalizable: both hold by Cor. 29. For potentially valuable/CBV solvable $\Longrightarrow$ typable: similar to Thm. 16, with the corresponding Lem. 31 and Lem. 32, and using the following two facts: (1) every value is typable and (2) `I` is typable with a solvable type. ◀

6 Extension to ΛJ_n, ΛJ_v and the λ-calculus

We have argued in favor of endowing generalized applications with a distant operational semantics: permutations are only used when they are necessary to unblock redexes, thus putting the focus on the computational content on the calculus, and also bringing the operational semantics of the calculus closer to the quantitative model. Nonetheless, this choice should not have an influence on overall properties such as strong normalization, solvability or potential valuability. We also wish to be conservative with respect to the original CBN and CBV calculi ΛJ_n and ΛJ_v. In this section we show that. More precisely, we prove the equivalence of CBN/CBV solvability with and without distance using the quantitative type systems introduced in previous sections. We also show that our CBN/CBV notion of solvability is equivalent to the original one for the λ-calculus, a result which is expected but not evident.

6.1 Solvability for ΛJ_n and ΛJ_v

Remember that $\rightarrow_{\mathtt{n}}$ (resp. $\rightarrow_{\mathtt{v}}$) is the reduction relation associated to the original CBN (resp. CBV) calculus. In what follows we write *local* to mean *non-distant*.

▶ **Definition 35** (Local Solvability). *Let $t \in \mathtt{T}_J$.*

(ΛJ_n) *t is **CBN local solvable** iff there is a head context $\mathtt{H}$ and a distant context $\mathtt{D}$ such that $\mathtt{H}\langle t\rangle \rightarrow^*_{\mathtt{n}} \mathtt{D}\langle \mathtt{I}\rangle$.*

(ΛJ_v) *t is **CBV local solvable** iff there is a head context $\mathtt{H}$ such that $\mathtt{H}\langle t\rangle \rightarrow^*_{\mathtt{v}} \mathtt{I}$.*

Notice that the terms $t_4 = \Omega(y, z.\mathtt{I})$ and $t_6 = x(\Omega, z.\mathtt{I})$ are CBN but not CBV locally solvable. The term $t_5 = y_1(\mathtt{I}, z_1.x)(y_2(\mathtt{I}, z_2.z_2), z_3.\lambda y.z_3)$ is both CBN and CBV solvable.

▶ **Definition 36.** *The **CBN local solvable reduction** $\rightarrow_{\mathtt{lsn}}$ is generated by the closure of the following rules $\beta\mathtt{h}$ and $\pi\mathtt{h}$ under head contexts.*

$$(\lambda x.t)(u, y.\mathtt{H}\langle\!\langle y\rangle\!\rangle) \mapsto_{\beta\mathtt{h}} \{\{u/x\}t/y\}\mathtt{H}\langle\!\langle y\rangle\!\rangle$$
$$t(u, x.r)(u', y.\mathtt{H}\langle\!\langle y\rangle\!\rangle) \mapsto_{\pi\mathtt{h}} t(u, x.r(u', y.\mathtt{H}\langle\!\langle y\rangle\!\rangle))$$

*The **local valuable reduction** $\rightarrow_{\mathtt{lpv}}$ and **CBV local solvable reduction** $\rightarrow_{\mathtt{lsv}}$ are defined by the closure of rules $\beta_{\mathtt{v}}$ and π under the same contexts used in their distant counterparts (Def. 19 and Def. 22 respectively).*

▶ **Theorem 37** (Local Characterization). *Let $t \in \mathtt{T}_J$. Then,*

CBN: *t is CBN local solvable iff t is $\cap N$-typable iff t is $\mathtt{lsn}$-normalizable.*

CBV: *t is CBV local potentially valuable iff t is $\cap V$-typable iff t is $\mathtt{lpv}$-normalizable, and t is CBV local solvable iff t is $\cap V$-typable with a solvable type iff t is $\mathtt{lsv}$-normalizable.*

Since the same notion of typability is used in the distant and local characterizations, this gives the following equivalence for free.

▶ **Corollary 38.** *CBN (resp. CBV) solvability is equivalent to CBN (resp. CBV) local solvability.*

6.2 Equivalence with Solvability in the λ-Calculus

We also relate solvability of generalized applications to solvability in the λ-calculus. More precisely, we consider λ-calculi with explicit substitutions. The set of terms with explicit substitutions $\mathtt{T}_{ES}$ is generated by the following grammar:

$$M, N ::= x \mid \lambda x.M \mid MN \mid [N/x]M$$

The last clause is an alternative notation for a let-binding `let` x `=` N `in` M.

We show that the following standard translations preserve solvability in both directions.

▶ **Definition 39** (Translations).

$$(\mathtt{T}_J \mapsto \mathtt{T}_{ES}) \quad x^* := x \quad (\lambda x.t)^* := \lambda x.t^* \quad t(u, x.r)^* := [t^* u^*/x]r^*$$
$$(\mathtt{T}_{ES} \mapsto \mathtt{T}_J) \quad x^\circ := x \quad (\lambda x.M)^\circ := \lambda x.M^\circ$$
$$(MN)^\circ := M^\circ(N^\circ, z.z) \quad ([N/x]M)^\circ := \mathtt{I}(N^\circ, x.M^\circ)$$

Our proofs use quantitative types. For CBN, we call $\mathcal{N}$ the type system in [28, 10], which characterizes head normalization and thus solvability. For CBV, we call $\mathcal{V}$ an alternative but faithful presentation of the type system in [5] for which all and only solvable terms are typable with a solvable type.

Indeed, we show that the translation of a $\mathtt{T}_J$-term typable in system $\cap N$ (resp. $\cap V$) is also typable in system $\mathcal{N}$ (resp. $\mathcal{V}$), and vice-versa.

▶ **Theorem 40** (Preservation of Typing). *Let $t \in \mathtt{T}_J$ and $M \in \mathtt{T}_{ES}$.*

- $\Gamma \Vdash_{\cap N} t : \sigma$ *implies* $\Gamma \Vdash_{\mathcal{N}} t^* : \sigma$ *and* $\Gamma \Vdash_{\mathcal{N}} M : \sigma$ *implies* $\Gamma \Vdash_{\cap N} M^\circ : \sigma$.
- $\Gamma \Vdash_{\cap V} t : \sigma$ *implies* $\Gamma \Vdash_{\mathcal{V}} t^* : \sigma$ *and* $\Gamma \Vdash_{\mathcal{V}} M : \sigma$ *implies* $\Gamma \Vdash_{\cap V} M^\circ : \sigma$.

As CBN/CBV solvability in the λ-calculus is equivalent to $\mathcal{N}$-typability/$\mathcal{V}$-typability with a solvable type, we get the final results:

▶ **Corollary 41.** *Let t be a $\mathtt{T}_J$-term.*

- *t is CBN solvable if and only if t^* is CBN solvable in the λ-calculus.*
- *t is CBV solvable if and only if t^* is CBV solvable in the λ-calculus.*

7 Related Works and Conclusion

Solvability for Plotkin's CBV calculus has been first studied in [32], where some key concepts (notably potentially valuable terms and solvable types) are introduced. Another contribution in this same framework is [20], where a genericity lemma is proved. An operational characterization of CBV solvability in terms of CBV reduction appears in [7], for a distant CBV calculus with explicit substitutions. In [12], a similar result is obtained for Plotkin's CBV λ-calculus combined with permutative rules used to unblock redexes. The authors of *op.cit.* also give a relational model of solvability, based on quantitative types. We draw inspiration from all these works to address the challenge of solvability in generalized applications, where stuck redexes are produced by the particular applicative structure of the calculus.

More generally, finding good operational formalisms for call-by-value is an active topic of research (see [4]), with new insights from linear logic [2, 22] and the sequent calculus [23]. We believe that λJ_v holds a singular place among them, thanks to its natural way to deal with stuck redexes. An open problem is to find a fully abstract model for the CBV λ-calculus. We would like to see whether generalized applications help in this quest. In particular, it would be interesting to understand CBV approximation for generalized applications, CBV Böhm trees [8, 26] based on the solvable reduction, as well as separability [31].

Idempotent intersection types for ΛJ_n are proposed in [30]. Although intersection types achieve to characterize all and only strongly normalizable terms, they do not reject π as a permutation rule for CBN, albeit quantitatively unsound. Quantitative types for CBN and λJ_n are introduced in [18], where only strong normalization has been addressed. Our quantitative CBN and CBV type systems can be seen as the first relational models for generalized applications. They are adapted from [10], but we take the key insight about solvable types from [5]. It would be interesting to see if the techniques developed for *tightness* [3, 29] can also be adapted to this framework.

References

1 Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *J. Funct. Program.*, 1(4):375–416, 1991. `doi:10.1017/S0956796800000186`.

2 Beniamino Accattoli. Proof nets and the call-by-value λ-calculus. *Theoretical Computer Science*, 606:2–24, November 2015. `doi:10.1016/j.tcs.2015.08.006`.

3 Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. Tight typings and split bounds, fully developed. *J. Funct. Program.*, 30:e14, 2020. doi:10.1017/S095679682000012X.

4 Beniamino Accattoli and Giulio Guerrieri. Open call-by-value. In *Programming Languages and Systems*, volume abs/1609.00322, pages 206–226. Springer International Publishing, 2016. doi:10.1007/978-3-319-47958-3_12.

5 Beniamino Accattoli and Giulio Guerrieri. Call-by-value solvability and multi types. *CoRR*, abs/2202.03079, 2022. arXiv:2202.03079.

6 Beniamino Accattoli and Delia Kesner. The structural λ-calculus. In *Computer Science Logic*, pages 381–395. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-15205-4_30.

7 Beniamino Accattoli and Luca Paolini. Call-by-value solvability, revisited. In *Functional and Logic Programming*, pages 4–16. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-29822-6_4.

8 Henk Pieter Barendregt. *The Lambda Calculus - Its Syntax and Semantics*. Elsevier, 1984. doi:10.1016/c2009-0-14341-6.

9 Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer Berlin Heidelberg, Lausanne, Switzerland, September 2007. doi:10.1007/978-3-540-74915-8_24.

10 Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, and Andrés Viso. The bang calculus revisited. In *Functional and Logic Programming*, pages 13–32. Springer International Publishing, 2020. doi:10.1007/978-3-030-59025-3_2.

11 Antonio Bucciarelli, Delia Kesner, and Daniel Ventura. Non-idempotent intersection types for the lambda-calculus. *Logic Journal of the IGPL*, 25(4):431–464, July 2017. doi:10.1093/jigpal/jzx018.

12 Alberto Carraro and Giulio Guerrieri. A semantical and operational account of call-by-value solvability. In *Lecture Notes in Computer Science*, volume 8412, pages 103–118. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-642-54830-7_7.

13 M. Coppo and M. Dezani-Ciancaglini. A new type assignment for λ-terms. *Archiv für Mathematische Logik und Grundlagenforschung*, 19(1):139–156, December 1978. doi:10.1007/bf02011875.

14 Daniel De Carvalho. Execution time of λ-terms via denotational semantics and intersection types. *Mathematical Structures in Computer Science*, 28(7):1169–1203, January 2017. doi:10.1017/s0960129516000396.

15 José Espírito Santo. Delayed substitutions. In *Lecture Notes in Computer Science*, pages 169–183. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-73449-9_14.

16 José Espírito Santo. The call-by-value lambda-calculus with generalized applications. In *28th EASCL Annual Conference on Computer Science Logic (CSL 2020)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2020. doi:10.4230/LIPICS.CSL.2020.35.

17 José Espírito Santo, Maria João Frade, and Luís Pinto. Permutability in proof terms for intuitionistic sequent calculus with cuts. In Silvia Ghilezan, Herman Geuvers, and Jelena Ivetić, editors, *22nd International Conference on Types for Proofs and Programs (TYPES 2016)*, volume 97 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:27, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TYPES.2016.10.

18 José Espírito Santo, Delia Kesner, and Loïc Peyrot. A faithful and quantitative notion of distant reduction for generalized applications. In *Proc. of the 24th Int. Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, LNCS, April 2022. arXiv:2201.04156.

19 José Espírito Santo and Luís Pinto. A calculus of multiary sequent terms. *ACM Transactions on Computational Logic*, 12(3):1–41, May 2011. doi:10.1145/1929954.1929959.

20 Álvaro García-Pérez and Pablo Nogueira. No solvable lambda-value term left behind. *Logical Methods in Computer Science*, 12(2), June 2016. doi:10.2168/lmcs-12(2:12)2016.

21 Philippa Gardner. Discovering needed reductions using type theory. In Masami Hagiya and John C. Mitchell, editors, *Lecture Notes in Computer Science*, pages 555–574. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994. doi:10.1007/3-540-57887-0_115.

22 Giulio Guerrieri, Luca Paolini, and Simona Ronchi Della Rocca. Standardization and conservativity of a refined call-by-value lambda-calculus. *Logical Methods in Computer Science ; Volume 13*, 2017. doi:10.23638/LMCS-13(4:29)2017.

23 Hugo Herbelin and Stéphane Zimmermann. An operational account of call-by-value minimal and classical λ-calculus in "natural deduction" form. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, pages 142–156, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-02273-9_12.

24 Felix Joachimski and Ralph Matthes. Standardization and confluence for a lambda calculus with generalized applications. In *Rewriting Techniques and Applications*, pages 141–155. Springer Berlin Heidelberg, 2000. doi:10.1007/10721975_10.

25 Felix Joachimski and Ralph Matthes. Short proofs of normalization for the simply-typed λ-calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic*, 42(1):59–87, 2003. doi:10.1007/s00153-002-0156-9.

26 Emma Kerinec, Giulio Manzonetto, and Michele Pagani. Revisiting Call-by-value Böhm trees in light of their Taylor expansion. *Logical Methods in Computer Science*, Volume 16, Issue 3, July 2020. doi:10.23638/LMCS-16(3:6)2020.

27 Delia Kesner. A theory of explicit substitutions with safe and full composition. *Log. Methods Comput. Sci.*, 5(3), 2009. arXiv:0905.2539.

28 Delia Kesner and Daniel Ventura. Quantitative types for the linear substitution calculus. In Josep Diaz, Ivan Lanese, and Davide Sangiorgi, editors, *Lecture Notes in Computer Science*, pages 296–310, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-44602-7_23.

29 Delia Kesner and Andrés Viso. Encoding tight typing in a unified framework. In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, volume 216 of *LIPIcs*, pages 27:1–27:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CSL.2022.27.

30 Ralph Matthes. Characterizing strongly normalizing terms for a lambda calculus with generalized applications via intersection types. In *Proc. ICALP 2000 (Geneva), volume 8 of Proceedings in Informatics*, pages 339–353, 2000.

31 Luca Paolini. Call-by-Value Separability and Computability. In Antonio Restivo, Simona Ronchi Della Rocca, and Luca Roversi, editors, *Theoretical Computer Science*, Lecture Notes in Computer Science, pages 74–89, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-45446-2_5.

32 Luca Paolini and Simona Ronchi Della Rocca. Call-by-value solvability. *RAIRO - Theoretical Informatics and Applications*, 33(6):507–534, November 1999. doi:10.1051/ita:1999130.

33 Gordon Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1:125–159, 1975. doi:10.1016/0304-3975(75)90017-1.

34 Neil Tennant. Ultimate normal forms for parallelized natural deductions. *Logic Journal of IGPL*, 10(3):299–337, 2002. doi:10.1093/jigpal/10.3.299.

35 Jan von Plato. Natural deduction with general elimination rules. *Archive for Mathematical Logic*, 40(7):541–567, 2001. doi:10.1007/s001530100091.

36 Christopher P. Wadsworth. The relation between computational and denotational properties for scott's D_∞-models of the lambda-calculus. *SIAM Journal on Computing*, 5(3):488–521, September 1976. doi:10.1137/0205036.

A Main proofs

In this section, we detail the crucial proofs showing that normalizability of a term t w.r.t. the CBN/CBV solvable reductions implies that t is CBN/CBV solvable. First, we define the number of head applications of a term as follows.

$$|x|_@ = 0 \qquad |\lambda x.t|_@ = |t|_@ \qquad |t(u, x.r)|_@ = \begin{cases} |r|_@ + |t|_@ + 1, & \text{if } x = \text{hv}(r) \\ |r|_@, & \text{otherwise} \end{cases}$$

Each proof will be decomposed in two statements. A first lemma states that solvable normal forms can be reduced to a value (surrounded by a distant context in the CBN case). The main property can then be shown by constructing an appropriate head context.

A.1 Call-by-Name

The following intermediate lemma is stated with β-reduction, instead of $\mathtt{d}\beta$-reduction. This enables us to use it in the characterizations of both distant and non-distant CBN.

▶ **Lemma 42.** *For all $t = \mathtt{H}\langle\langle x\rangle\rangle \in \text{NF}_{\mathtt{sn}}$, $n \geq |t|_@$ and distant context $\mathtt{D}_0$, there is $m \geq 0$, there are variables $x_1, \ldots, x_m$ and distant contexts $\mathtt{D}, \mathtt{D}_1, \ldots \mathtt{D}_m$ such that $\{\mathtt{D}_0\langle \mathbf{o}^n\rangle/x\}t \rightarrow^*_\beta \mathtt{D}\langle\lambda x_m.\mathtt{D}_m\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|t|_@}\rangle\rangle\rangle$. In particular, if t is neutral normal, then $m = 0$.*

Proof. By induction on $\langle |t|_@, t\rangle$. We reason by cases on the form of the normal term t.

- $t = x$, so that $|t|_@ = 0$ (this is the base case of the induction). We let $m = 0$, $\mathtt{D} = \mathtt{D}_0$ and conclude since $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}x = \mathtt{D}\langle\mathbf{o}^n\rangle = \mathtt{D}\langle\mathbf{o}^{n-|x|_@}\rangle$.
- $t = \lambda y.t'$, where $t' = \mathtt{H}'\langle\langle x\rangle\rangle$ ($x \neq y$). We suppose w.l.o.g that $y \notin \text{fv}(\mathtt{D}_0\langle\mathbf{o}^n\rangle)$. Let $n \geq |t|_@ = |t'|_@$. By the *i.h.* there are m', $x_1, \ldots, x_{m'}$ and $\mathtt{D}', \mathtt{D}_1, \ldots, \mathtt{D}_{m'}$ such that $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}t' \rightarrow^*_\beta \mathtt{D}'\langle\lambda x_{m'}.\mathtt{D}_{m'}\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|t'|_@}\rangle\rangle\rangle$. Thus we obtain $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}t \rightarrow^*_\beta \lambda y.\mathtt{D}'\langle\lambda x_{m'}.\mathtt{D}_{m'}\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|t'|_@}\rangle\rangle\rangle$ since $|t|_@ = |t'|_@$. We conclude by taking $m = m' + 1$, $x_m = y$, $\mathtt{D}_m = \mathtt{D}'$ and $\mathtt{D} = \Diamond$.
- $t = s(u, y.r)$, where $r = \mathtt{H}'\langle\langle y\rangle\rangle$ ($y \neq x$) and $s = \mathtt{G}\langle\langle x\rangle\rangle$. We have $|t|_@ = |s|_@ + |r|_@ + 1$. Let $n \geq |t|_@ > |s|_@$. Applying the *i.h.* on the neutral normal term s, we know that for any $\mathtt{D}_0$, there is $\mathtt{D}'$ such that $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}s \rightarrow^*_\beta \mathtt{D}'\langle\mathbf{o}^{n-|s|_@}\rangle$.
  Let $n' = n - |s|_@ - 1$. Then $n' \geq |r|_@$ and we can show that $\{\mathtt{D}_0\langle\mathbf{o}^{n'+1}\rangle/x\}r \in \text{NF}_{\mathtt{sn}}$ and $|\{\mathtt{D}_0\langle\mathbf{o}^{n'+1}\rangle/x\}r|_@ = |r|_@$. Moreover, $\{\mathtt{D}_0\langle\mathbf{o}^{n'+1}\rangle/x\}r$ is of the form $\mathtt{H}''\langle\langle y\rangle\rangle$, for some $\mathtt{H}''$. We can then apply the *i.h.* on $\{\mathtt{D}_0\langle\mathbf{o}^{n'+1}\rangle/x\}r$, so there are $m', x_1, \ldots, x_{m'}, \mathtt{D}, \mathtt{D}_1, \ldots, \mathtt{D}_{m'}$ such that $\{\mathtt{D}'\langle\mathbf{o}^{n'}\rangle/y\}\{\mathtt{D}_0\langle\mathbf{o}^{n'+1}\rangle/x\}r \rightarrow^*_\beta \mathtt{D}\langle\lambda x_{m'}.\mathtt{D}_{m'}\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n'-|r|_@}\rangle\rangle\rangle$. We take $m = m'$. In the case where t is neutral normal, we have $m = 0$ as required. Since $n' - |r|_@ = n - |t|_@$, we conclude as follows:

$$\begin{aligned}\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}t &= \{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}s(\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}u, y.\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}r) \\ &\rightarrow^*_\beta \mathtt{D}'\langle\mathbf{o}^{n'+1}\rangle(\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}u, y.\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}r) \\ &\rightarrow_\beta \{\mathtt{D}'\langle\mathbf{o}^{n'}\rangle/y\}\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}r \rightarrow^*_\beta \mathtt{D}\langle\lambda x_m.\mathtt{D}_m\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|t|_@}\rangle\rangle\rangle.\end{aligned}$$

- $t = s(u, y.t')$, where $y \neq \text{hv}(t')$. Let $n \geq |t|_@ = |t'|_@$. By the *i.h.* on t', for all $\mathtt{D}_0$ there are $m', x_1, \ldots, x_{m'}, \mathtt{D}', \mathtt{D}_1, \ldots, \mathtt{D}_{m'}$ s.t. $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}t' \rightarrow^*_\beta \mathtt{D}'\langle\lambda x_{m'}.\mathtt{D}_{m'}\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|t'|_@}\rangle\rangle\rangle$. In particular $m' = 0$ if t' is neutral normal. We set $\mathtt{D} = \{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}s(\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}u, y.\mathtt{D}')$ and $m = m'$. Since $|t'|_@ = |t|_@$, then $\{\mathtt{D}_0\langle\mathbf{o}^n\rangle/x\}t \rightarrow^*_\beta \mathtt{D}\langle\lambda x_m.\mathtt{D}_m\langle\ldots\lambda x_1.\mathtt{D}_1\langle\mathbf{o}^{n-|s|_@}\rangle\rangle\rangle$. ◀

▶ **Lemma 6.** *Let t be an* sn*-normalizable term. Then t is CBN solvable.*

Proof. Since t is sn-normalizable, then there is a solvable normal term $t' \in \mathrm{NF}_{\mathtt{sn}}$ such that $t \to_{\mathtt{sn}}^{*} t'$ (and thus $t \to_{\mathtt{dn}}^{*} t'$). Let $\mathrm{hv}(t') = x$. The term t' can take two shapes:

1. $t' = \mathtt{H}'\langle\!\langle x \rangle\!\rangle$ if $x \in \mathrm{fv}(t')$;
2. $t' = \mathtt{D}_l\langle \lambda x_l. \ldots \lambda x_2.\mathtt{D}_1\langle \lambda x_1.\mathtt{H}'\langle\!\langle x \rangle\!\rangle\rangle\rangle$ for $x \in \{x_1, \ldots, x_l\}$, if $x \notin \mathrm{fv}(t')$.

In both cases, we must give a head context $\mathtt{H}$ such that $\mathtt{H}\langle t \rangle \to_{\mathtt{dn}}^{*} \mathtt{D}\langle \mathtt{I} \rangle$ for a distant context $\mathtt{D}$.

We start with the first case (x is free in t'). Let $n = |t'|_{@}$. By Lem. 42, there are $m \geq 0$, variables $y_1, \ldots, y_m$ and distant contexts $\mathtt{D}', \mathtt{D}_1, \ldots, \mathtt{D}_m$ such that $\{\mathbf{o}^n/x\}t' \to_{\beta}^{*} \mathtt{D}'\langle \lambda y_m.\mathtt{D}_m\langle \ldots \lambda y_1.\mathtt{D}_1\langle \mathtt{I} \rangle\rangle\rangle$, which is also a dn-step. We let $\mathtt{H} = (\lambda x.\Diamond)(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m}$. Then, we have:

$$\begin{aligned}\mathtt{H}\langle t \rangle \to_{\mathtt{dn}}^{*} \mathtt{H}\langle t' \rangle = (\lambda x.t')(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m} &\to_{\mathtt{dn}} \{\mathbf{o}^n/x\}t'\overline{(\mathtt{I}, z.z)}^{m} \\ \to_{\mathtt{dn}}^{*} \mathtt{D}'\langle \lambda y_m.\mathtt{D}_m\langle \ldots \lambda y_1.\mathtt{D}_1\langle \mathtt{I} \rangle\rangle\rangle\overline{(\mathtt{I}, z.z)}^{m} &\to_{\mathtt{dn}}^{m} \mathtt{D}'\langle\{\mathtt{I}/y_m\}\mathtt{D}_m\langle \ldots \{\mathtt{I}/y_1\}\mathtt{D}_1\langle \mathtt{I} \rangle\rangle\rangle\end{aligned}$$

We conclude by taking $\mathtt{D} = \mathtt{D}'\langle\{\mathtt{I}/y_m\}\mathtt{D}_m\langle \ldots \{\mathtt{I}/y_1\}\mathtt{D}_1\rangle\rangle$.

In the second case (x is not free in t'), let $1 \leq i \leq l$ such that $x = x_i$. Let us consider the following reduction sequence:

$$t'\overline{(\mathtt{I}, z.z)}^{l-i} = \mathtt{D}_l\langle \lambda x_l. \ldots \mathtt{D}_1\langle \lambda x_1.\mathtt{H}'\langle\!\langle x \rangle\!\rangle\rangle\rangle\overline{(\mathtt{I}, z.z)}^{l-i} \to_{\mathtt{dn}}^{l-i} \mathtt{D}''\langle \lambda x.\mathtt{H}''\langle\!\langle x \rangle\!\rangle\rangle$$

where $\mathtt{D}'' = \mathtt{D}_l\langle\{\mathtt{I}/x_l\}\mathtt{D}_{l-1}\langle \ldots \{\mathtt{I}/x_{i+1}\}\mathtt{D}_i\rangle\rangle$ and $\mathtt{H}'' = \{\mathtt{I}/x_j\}_{i<j\leq l}\mathtt{D}_{i-1}\langle \lambda x_{i-1}. \ldots \mathtt{D}_1\langle \lambda x_1.\mathtt{H}'\rangle\rangle$. The subterm $\mathtt{H}''\langle\!\langle x \rangle\!\rangle$ above is obtained by substituting a sn-normal term with variables different from the head variable. This kind of substitution preserves the property of being sn-normal, so that $\mathtt{H}''\langle\!\langle x \rangle\!\rangle$ is sn-normal.

Let $n = |\mathtt{H}''\langle\!\langle x \rangle\!\rangle|_{@}$. Then Lem. 42 applied to $\{\mathbf{o}^n/x\}\mathtt{H}''\langle\!\langle x \rangle\!\rangle$ gives integers $m, y_1, \ldots, y_m$ and distant contexts $\mathtt{D}', \mathtt{D}'_1, \ldots, \mathtt{D}'_m$ such that (this is also a dn-step):

$$\{\mathbf{o}^n/x\}\mathtt{H}''\langle\!\langle x \rangle\!\rangle \to_{\beta}^{*} \mathtt{D}'\langle \lambda y_m.\mathtt{D}'_m\langle \ldots \lambda y_1.\mathtt{D}'_1\langle \mathtt{I} \rangle\rangle\rangle.$$

To conclude, we let $\mathtt{H} = \Diamond\overline{(\mathtt{I}, z.z)}^{l-i}(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m}$, where m and n were obtained before. The whole reduction from $\mathtt{H}\langle t \rangle$ goes as follows:

$$\begin{aligned}\mathtt{H}\langle t \rangle \to_{\mathtt{dn}}^{*} \mathtt{H}\langle t' \rangle &= t'\overline{(\mathtt{I}, z.z)}^{l-i}(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m} \\ &= \mathtt{D}_l\langle \lambda x_l. \ldots \mathtt{D}_1\langle \lambda x_1.\mathtt{H}'\langle\!\langle x \rangle\!\rangle\rangle\rangle\overline{(\mathtt{I}, z.z)}^{l-i}(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m} \\ &\to_{\mathtt{dn}}^{l-i} \mathtt{D}''\langle \lambda x.\mathtt{H}''\langle\!\langle x \rangle\!\rangle\rangle(\mathbf{o}^n, z.z)\overline{(\mathtt{I}, z.z)}^{m} \\ &\to_{\mathtt{dn}} \mathtt{D}''\langle\{\mathbf{o}^n/x\}\mathtt{H}''\langle\!\langle x \rangle\!\rangle\rangle\overline{(\mathtt{I}, z.z)}^{m} \\ &\to_{\mathtt{dn}}^{*} \mathtt{D}''\langle \mathtt{D}'\langle \lambda y_m.\mathtt{D}'_m\langle \ldots \lambda y_1.\mathtt{D}'_1\langle \mathtt{I} \rangle\rangle\rangle\rangle\overline{(\mathtt{I}, z.z)}^{m} \\ &\to_{\mathtt{dn}}^{m} \mathtt{D}''\langle \mathtt{D}'\langle\{\mathtt{I}/y_m\}\mathtt{D}'_m\langle \ldots \{\mathtt{I}/y_1\}\mathtt{D}'_1\langle \mathtt{I} \rangle\rangle\rangle\rangle\end{aligned}$$

where $\mathtt{D}'' = \mathtt{D}_l\langle\{\mathtt{I}/x_l\}\mathtt{D}_{l-1}\langle \ldots \{\mathtt{I}/x_{i+1}\}\mathtt{D}_i\rangle\rangle$ and $\mathtt{H}'' = \{\mathtt{I}/x_j\}_{i<j\leq l}\mathtt{D}_{i-1}\langle \lambda x_{i-1}. \ldots \mathtt{D}_1\langle \lambda x_1.\mathtt{H}'\rangle\rangle$.

We conclude by taking $\mathtt{D} = \mathtt{D}''\langle \mathtt{D}'\langle\{\mathtt{I}/y_m\}\mathtt{D}'_m\langle \ldots \{\mathtt{I}/y_1\}\mathtt{D}'_1\rangle\rangle\rangle$ so that $\mathtt{H}\langle t \rangle \to_{\mathtt{dn}}^{*} \mathtt{D}\langle \mathtt{I} \rangle$. ◀

A.2 Call-by-Value

Before showing the property that sv-normalizable terms are CBV solvable, we need to show that pv-normalizable terms are potentially valuable. Here also, we use an intermediate lemma to prove that pv-nfs can be reduced to a value. The main difference between this lemma, as well as the one for sv, and the corresponding one in CBN, is that we must assign arbitrary terms $\mathbf{o}^n$ to free variables of t, and not just to the head variable. This lemma proceeds by induction on the grammar $\mathrm{NF}_{\mathtt{pv}}$, we do not give the proof for lack of space.

▶ **Lemma 43.** *For all $t \in \mathrm{NF}_{\mathtt{pv}}$ with $\mathrm{fv}(t) \subseteq \{x_1, \ldots, x_m\}$, there exists $h \geq |t|_@$ such that for all $n_1, \ldots, n_m \geq h$ there exists a value v such that $\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}t \to^*_{\beta_\mathtt{v}} v$. If $t \in \mathrm{NE}_{\mathtt{pv}}$ with $\mathrm{hv}(t) = x_i$ (necessarily free), then $v = \mathbf{o}^{n_i - |t|_@}$.*

▶ **Lemma 21.** *Let t be a* $\mathtt{pv}$*-normalizable term. Then t is potentially valuable.*

Proof. Since t is $\mathtt{pv}$-normalizable, then there is a $\mathtt{pv}$-normal term t' such that $t \to^*_{\mathtt{pv}} t'$. Therefore $t' \in \mathrm{NF}_{\mathtt{pv}}$ by Lem. 20. Let $\mathrm{fv}(t) = \{x_1, \ldots, x_m\}$, so that $\mathrm{fv}(t') \subseteq \{x_1, \ldots, x_m\}$. By Lem. 43, there is $h \geq |t'|_@$ such that $\{\mathbf{o}^h/x_m\} \ldots \{\mathbf{o}^h/x_1\}t' \to^*_{\beta_\mathtt{v}} v$ for some value v. Consider $\mathtt{D} = \mathtt{I}(\mathbf{o}^h, x_1.\mathtt{I}(\mathbf{o}^h, x_2. \ldots \mathtt{I}(\mathbf{o}^h, x_m.\Diamond) \ldots))$. Then,

$$\begin{aligned} \mathtt{D}\langle t \rangle &\to^*_{\mathtt{pv}} \mathtt{I}(\mathbf{o}^h, x_1. \ldots \mathtt{I}(\mathbf{o}^h, x_m.t')) \to_{\beta_\mathtt{v}} \mathtt{I}(\mathbf{o}^h, x_2. \ldots \mathtt{I}(\mathbf{o}^h, x_m.\{\mathbf{o}^h/x_1\}t') \ldots) \\ &\to^*_{\beta_\mathtt{v}} \{\mathbf{o}^h/x_m\} \ldots \{\mathbf{o}^h/x_1\}t' \to^*_{\beta_\mathtt{v}} v \text{ (by Lem. 43)} \end{aligned}$$

As a consequence, $\mathtt{D}\langle t \rangle \to^*_{\mathtt{dv}} v$. ◀

We are now ready to prove the property for $\mathtt{sv}$-reduction. First, the intermediate Lemma.

▶ **Lemma 44.** *For all $t \in \mathrm{NF}_{\mathtt{sv}}$ with $\mathrm{fv}(t) \subseteq \{x_1, \ldots, x_m\}$, there exist $h \geq |t|_@, k \geq 0$ such that for all $n_1, \ldots, n_{m+k} \geq h$ there exists $n \geq 0$ such that the following holds $\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}t(\mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n$.*

Proof. By induction on $t \in \mathrm{NF}_{\mathtt{sv}}$.

- t is a variable, thus $t = x_i$. We take $h = 0 = |x_i|_@, k = 0$ so that for all $n_1, \ldots, n_m \geq 0$ we have $\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}t = \mathbf{o}^{n_i}$. We let $n = n_i \geq 0$ and we conclude.
- $t = \lambda x.s$ with $s \in \mathrm{NF}_{\mathtt{sv}}$. We suppose w.l.o.g that $x \notin \{x_1, \ldots, x_m\}$. Then, $\mathrm{fv}(s) \subseteq \{x, x_1, \ldots, x_m\}$. By the *i.h.*, there exist $h' \geq |s|_@ = |t|_@, k' \geq 0$ such that for all $n', n_1, \ldots, n_{m+k} \geq h'$ there exists $n \geq 0$ such that

  $$\{\mathbf{o}^{n'}/x\}\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}s(\mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n\,.$$

  Taking $h = h'$ and $k = k' + 1$ we have:

  $$\begin{aligned} &\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}t(\mathbf{o}^{n'}, \mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \\ &= \lambda x.\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}s(\mathbf{o}^{n'}, \mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \\ &\to_{\beta_\mathtt{v}} \{\mathbf{o}^{n'} \backslash\!\backslash x\}\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}s(\mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \\ &= \{\mathbf{o}^{n'}/x\}\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}s(\mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n \ \text{(by the } \textit{i.h.}) \end{aligned}$$

- $t = s(u, y.r)$ with $s \in \mathrm{NE}_{\mathtt{pv}}$, $u \in \mathrm{NF}_{\mathtt{pv}}$ and $r \in \mathrm{NF}_{\mathtt{sv}}$. We suppose w.l.o.g that $y \notin \{x_1, \ldots, x_m\}$. Thus $\mathrm{fv}(r) \subseteq \{y, x_1, \ldots, x_m\}$. Let $x_j = \mathrm{hv}(s)$ for some $1 \leq j \leq m$. By Lem. 43 and the *i.h.*respectively:
  1. There is $h_s \geq |s|_@$ s.t. for all $n^s_1, \ldots, n^s_m \geq h_s$ we have $\{\mathbf{o}^{n^s_m}/x_m\} \ldots \{\mathbf{o}^{n^s_1}/x_1\}s \to^*_{\beta_\mathtt{v}} \mathbf{o}^{n^s_j - |s|_@}$.
  2. There is $h_u \geq |u|_@$ such that for all $n^u_1, \ldots, n^u_m \geq h_u$ there is a value v such that $\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}u \to^*_{\beta_\mathtt{v}} v$.
  3. There are $h_r \geq |r|_@, k' \geq 0$ such that for all $n_y, n^r_1, \ldots, n^r_{m+k'} \geq h_r$ there is $n \geq 0$ such that $\{\mathbf{o}^{n_y}/y\}\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}r(\mathbf{o}^{n_{m+1}}, \ldots, \mathbf{o}^{n_{m+k'}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n$.

  We take $h = \max(h_s + h_r + 1, h_u) \geq |t|_@$ and we consider any $n_1, \ldots, n_m \geq h$.
  - We have $h \geq h_s + h_r + 1$ and thus $n_1, \ldots, n_m \geq h$ implies in particular $n_1, \ldots, n_m \geq h_s$. This gives $\{\mathbf{o}^{n_m}/x_m\} \ldots \{\mathbf{o}^{n_1}/x_1\}s \to^*_{\beta_\mathtt{v}} \mathbf{o}^{n_j - |s|_@}$ by (1).

- We have $h \geq h_u$ and thus $n_1, \dots, n_m \geq h$ implies in particular $n_1, \dots, n_m \geq h_u$. This gives $\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} u \to^*_{\mathtt{pv}} v$ by (2).
- We have $h \geq h_r + 1 > h_r$ and thus $n_1, \dots, n_m \geq h$ implies in particular $n_1, \dots, n_m \geq h_r + h_s + 1 \geq h_r + |s|_@ + 1 > h_r$. This gives $n \geq 0$ such that by the *i.h.* (3) $\{\mathbf{o}^{n_j - |s|_@ - 1} / y\}\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} r(\mathbf{o}^{n_{m+1}}, \dots, \mathbf{o}^{n_{m+k'}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n$.

In summary, we reduce as follows:

$$\begin{aligned}
&\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} t(\mathbf{o}^{n_{m+1}}, \dots, \mathbf{o}^{n_{m+k}}, z.z) \\
&\to^*_{\beta_\mathtt{v}} \mathbf{o}^{n_j - |s|_@}(v, y.\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} r)(\mathbf{o}^{n_{m+1}}, \dots, \mathbf{o}^{n_{m+k}}, z.z) \\
&\to_{\beta_\mathtt{v}} \{\mathbf{o}^{n_j - |s|_@ - 1} / y\}\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} r(\mathbf{o}^{n_{m+1}}, \dots, \mathbf{o}^{n_{m+k}}, z.z) \\
&\to^*_{\beta_\mathtt{v}} \mathbf{o}^n \;\; \text{(by the } \textit{i.h.} \text{ (3))}
\end{aligned}$$ ◀

▶ **Lemma 24.** *Let t be an $\mathtt{sv}$-normalizable term. Then t is CBV solvable.*

Proof. Since t is $\mathtt{sv}$-normalizable, then there is an $\mathtt{sv}$-normal term t' such that $t \to^*_{\mathtt{sv}} t'$. Therefore $t' \in \mathrm{NF}_{\mathtt{sv}}$ by Lem. 23. Let $\mathrm{fv}(t) = \{x_1, \dots, x_m\}$, so that $\mathrm{fv}(t') \subseteq \{x_1, \dots, x_m\}$. By Lem. 42, there are $h, k \in \mathbb{N}$ such that for all $n_1, \dots, n_{m+k} \geq h$ there is $n \geq 0$ such that $\{\mathbf{o}^{n_m} / x_m\} \dots \{\mathbf{o}^{n_1} / x_1\} t'(\mathbf{o}^{n_{m+1}}, \dots, \mathbf{o}^{n_{m+k}}, z.z) \to^*_{\beta_\mathtt{v}} \mathbf{o}^n$, which is also a $\mathtt{dv}$-step. We take $n_1, \dots, n_{m+k} = h$. We can then write $(\mathbf{o}^h, \dots, \mathbf{o}^h, z.z)$ as $\overline{(\mathbf{o}^h, z.z)}^k$. Let $\mathtt{H} = \mathtt{I}(\mathbf{o}^h, x_m. \dots \mathtt{I}(\mathbf{o}^h, x_1.\Diamond) \dots)\overline{(\mathbf{o}^h, z.z)}^k\overline{(\mathtt{I}, z.z)}^n$. Then:

$$\mathtt{H}\langle t\rangle \to^*_{\mathtt{sv}} \mathtt{H}\langle t'\rangle \to^m_{\beta_\mathtt{v}} \mathtt{dv}\{\mathbf{o}^h / x_m\} \dots \{\mathbf{o}^h / x_1\} t'\overline{(\mathbf{o}^h, z.z)}^k\overline{(\mathtt{I}, z.z)}^n \to^*_{\beta_\mathtt{v}} \mathbf{o}^n\,\overline{(\mathtt{I}, z.z)}^n \to^n_{\beta_\mathtt{v}} \mathtt{I}$$

As a consequence, $\mathtt{H}\langle t\rangle \to_{\mathtt{dv}} \mathtt{I}$. ◀

B Quantitative Type Systems for λ-calculi with Explicit Substitutions

We use the following type system for the proofs of equivalence between our calculi and the λ-calculus with explicit substitutions in Sec. 6.

We start with the **CBN type system** of [28]. The grammar of types is the same as ours for the CBN system. This system differs, in the presentation only, by the fact that we write (MANY) as a separate rule.

$$\frac{}{x : [\sigma] \vdash x : \sigma}\,(\textsc{ax}) \qquad \frac{(\Gamma_i \vdash M : \sigma_i)}{\wedge_{i \in I}\Gamma_i \vdash M : [\sigma_i]_{i \in I}}\,(\textsc{many}) \qquad \frac{\Gamma; x : \mathcal{M} \vdash M : \sigma}{\Gamma \vdash \lambda x.M : \mathcal{M} \to \sigma}\,(\to_i)$$

$$\frac{\Gamma \vdash M : \mathcal{M} \to \sigma \qquad \Delta \vdash N : \mathcal{M}}{\Gamma \wedge \Delta \vdash MN : \sigma}\,(\to_e) \qquad \frac{\Gamma; x : \mathcal{M} \vdash M : \sigma \qquad \Delta \vdash N : \mathcal{M}}{\Gamma \wedge \Delta \vdash [N/x]M : \sigma}\,(\textsc{cut})$$

For the **CBV type system**, we define a new type system using the rules of the system from [5], but using our CBV type grammar. A difference in the presentation is that we include rule (MANY) inside the rules for variables and abstractions. Typability is equivalent in both systems.

$$\frac{}{x : \mathcal{M} \vdash x : \mathcal{M}}\,(\textsc{ax}) \qquad \frac{(\Gamma_i; x : \mathcal{M}_i \vdash M : \sigma_i)_{i \in I}}{\wedge_{i \in I}\Gamma_i \vdash \lambda x.M : [\mathcal{M}_i \to \sigma_i]_{i \in I}}\,(\lambda)$$

$$\frac{\Gamma \vdash M : [\mathcal{M} \to \mathcal{N}] \qquad \Delta \vdash N : \mathcal{M}}{\Gamma \wedge \Delta \vdash MN : \mathcal{N}}\,(@) \qquad \frac{\Gamma; x : \mathcal{M} \vdash M : \sigma \qquad \Delta \vdash N : \mathcal{M}}{\Gamma \wedge \Delta \vdash [N/x]M : \sigma}\,(\textsc{es})$$