



HAL
open science

Strong Call-by-Value and Multi Types

Beniamino Accattoli, Giulio Guerrieri, Maico Leberle

► **To cite this version:**

Beniamino Accattoli, Giulio Guerrieri, Maico Leberle. Strong Call-by-Value and Multi Types. ICTAC 2023 - 20th International Colloquium on Theoretical Aspects of Computing, Dec 2023, Lima, Peru. hal-04395549

HAL Id: hal-04395549

<https://hal.science/hal-04395549v1>

Submitted on 15 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Strong Call-by-Value and Multi Types

Beniamino Accattoli¹[0000–0003–4944–9944], Giulio Guerrieri²[0000–0002–0469–4279], and Maico Leberle¹

¹ Inria & LIX, École Polytechnique, UMR 7161, Palaiseau, France
beniamino.accattoli@inria.fr, maico-carlos.leberle@inria.fr

² Aix Marseille Univ, CNRS, LIS UMR 7020, Marseille, France
giulio.guerrieri@lis-lab.fr

Abstract. This paper provides foundations for strong (that is, possibly under abstraction) call-by-value evaluation for the λ -calculus. Recently, Accattoli et al. proposed a form of call-by-value strong evaluation for the λ -calculus, the *external strategy*, and proved it reasonable for time. Here, we study the external strategy using a semantical tool, namely Ehrhard’s call-by-value multi types, a variant of intersection types. We show that the external strategy terminates exactly when a term is typable with so-called *shrinking multi types*, mimicking similar results for strong call-by-name. Additionally, the external strategy is normalizing in the untyped setting, that is, it reaches the normal form whenever it exists. We also consider the *call-by-extended-value* approach to strong evaluation shown reasonable for time by Biernacka et al. The two approaches turn out to *not* be equivalent: terms may be externally divergent but terminating for call-by-extended-value.

1 Introduction

Plotkin’s call-by-value λ -calculus λ_v [40] is at the heart of programming languages such as OCaml and proof assistants such as Coq. In the study of programming languages, call-by-value (shortened to CbV) evaluation is usually *weak*, that is, it does not reduce under abstractions, and terms are assumed to be *closed*, *i.e.*, without free variables. These constraints give rise to an elegant framework—we call it *Closed CbV*, following Accattoli and Guerrieri [5].

Plotkin did not present the CbV λ -calculus λ_v with these restrictions, and properties such as confluence also hold without the restrictions. As soon as open terms are allowed, however, or evaluation is *strong* (that is, it can reduce under abstractions), the calculus behaves badly at the semantical level. There are at least two issues, first pointed out by Paolini and Ronchi Della Rocca [39,38,41].

1. *False normal forms*: some terms are contextually equivalent to the looping term $\Omega := (\lambda x.xx)(\lambda x.xx)$ and yet they are normal in Plotkin’s setting.
2. *Failing of denotational soundness/adequacy beyond the closed case*: denotational models are usually both *sound* (that is, denotations are stable by reduction: if $t \rightarrow u$ then $\llbracket t \rrbracket = \llbracket u \rrbracket$) and *adequate* (that is, the denotation $\llbracket t \rrbracket$ is non-empty if and only if the evaluation of t terminates) only for Closed CbV; at least one of the two properties fails in the open/strong case.

Extensions of Plotkin’s Call-by-Value. A number of calculi extending Plotkin’s λ_v have been proposed. A first line of work studies a related and yet different issue of λ_v , namely the equational incompleteness with respect to continuation-passing translations, pointed out by Plotkin himself in [40]. This issue was solved with categorical tools by Moggi [37], which led to a number of studies, among others [42,43,34,21,24,30], that introduced many proposals of improved calculi for CbV.

A second and more recent line of work, due to Accattoli, Guerrieri, and coauthors, addresses the problem of open terms and strong evaluation directly [11,17,1,5,29,6,7,2,8]. It builds on the work of Paolini and Ronchi Della Rocca and on tools and techniques coming from the theory of Girard’s linear logic [26].

In [5], they compare four different extensions of Plotkin’s calculus in the framework of weak evaluation with possibly open terms. Their result is that the four calculi are all *termination equivalent*: t terminates in one of these extensions if and only if terminates in the other ones. In particular, in these extensions the issue of *false normal forms* is solved because all terms contextually equivalent to Ω do diverge, in contrast to what happens in Plotkin’s calculus λ_v . The notion of termination shared by the four calculi is then referred to as *Open CbV* in [5].

One of the aims of this paper is identifying an analogous notion of termination for *strong* CbV evaluation. Perhaps surprisingly, indeed, the termination equivalent calculi of Open CbV do not agree on what such a notion should be.

Two Relevant Extensions. Two Open CbV calculi are relevant here. The first one is a *call-by-extended-values* λ -calculus where the restriction on β -redexes *by value* is weakened to β -redexes having as argument an extended, more general notion of *value*. First used as a nameless technical tool by Paolini and Ronchi Della Rocca [39,41], then rediscovered by Accattoli and Sacerdoti Coen [12] to study cost models, it has some similarities with a calculus introduced by Grégoire and Leroy [27] to study a CbV abstract machine for Coq. In [12], extended values are called *fireballs* (a pun on *fire-able*) and the calculus is called *fireball calculus*.

The second extension is the *value substitution calculus* (shortened to VSC) due to Accattoli and Paolini and related to linear logic proof nets [11,1]. It was introduced to overcome some of the semantical problems of Plotkin’s setting, and it is a flexible tool, used in particular to relate the four extensions in [5].

Beyond False Normal Forms. In later works [6,8], Accattoli and Guerrieri show that the termination equivalence of Open CbV does not necessarily solve the other semantical issue of Open CbV, namely *the failing of denotational soundness/adequacy beyond the closed case*. On the one hand, they show that the fireball calculus is adequate but *not sound* with respect to Ehrhard’s CbV relational model [25], a paradigmatic model arising from the theory of linear logic and handily presented as a *multi types system* (a variant of intersection types). On the other hand, they show that the open VSC is *both* adequate and sound with respect to that model, suggesting that it is a better setting for Open CbV.

Strong Call-by-Value. The strong case has received less attention. In particular, it is not even clear what is the right notion of termination. The recent literature

contains two proposals of strong CbV evaluation, which have been carefully studied from the point of views of abstract machines and reasonable cost models, but not from a semantical point of view. The first one is the *strong fireball calculus*, for which abstract machines have been recently designed in 2020 [14] and 2021 [15] by Biernacka et al., the latter being reasonable for time (defined as the number of β -steps). The second proposal is the strong VSC, and more precisely the *external strategy* of the strong VSC, introduced in 2021 by Accattoli et al. [2], together with a reasonable machine implementing it.

The works [15] and [2] have been developed independently and at the same time, by two distinct groups, who cite each other. They state that both implement *Strong CbV*, but they fail to notice that they implement *different notions of termination*, raising the question of what exactly should be considered as Strong CbV. As we point out here, indeed, some terms are normalizing in the strong fireball calculus but have no normal form with respect to the external strategy.

Strong Call-by-Value and Multi Types. To clarify the situation, we here explore the semantic perspective provided by CbV multi types. For such types, typability coincides with termination of *open* CbV evaluation, as shown by Accattoli and Guerrieri [28,6,8], so they do not directly model strong evaluation. A similar mismatch happens in call-by-name (CbN for short), where terms typable with multi types coincide with the *head* (rather than strong) terminating ones. It is well known, however, that the restriction to *shrinking types* (that have no negative occurrences of the empty multiset) does model strong evaluation: in CbN, terms typable with shrinking types coincide with the leftmost(-outermost) terminating ones, and the leftmost strategy is a normalizing strategy of Strong CbN. Such a use of shrinkingness is standard in the theory of intersection and multi types, see Krivine [32], de Carvalho [19], Kesner and Ventura [31], Bucciarelli et al. [16].

Here, we adapt to CbV the shrinking technique as presented by Accattoli et al. for CbN multi types in [4], where the *shrinking* terminology is also introduced. Our main result is the characterization of external termination via shrinking types: a term t is typable with shrinking CbV multi types *if and only if* the external strategy terminates on t . Technically, the result is a smooth adaptation of the technique in [4]. Smoothness is here a *plus*, as it shows that the external strategy is the notion of Strong CbV termination *naturally* validated by CbV multi types, without ad-hoc stretchings of the technique.

Untyped Normalization Theorem. In an untyped setting, not every term normalizes (think of Ω) and in the strong case some terms have both reductions that normalize and reductions that diverge, for instance $(\lambda x.y)(\lambda z.\Omega)$. Thus, it is important to have a strategy that reaches a normal form whenever possible, i.e., that is *normalizing* in an untyped setting. The canonical evaluation strategy in Strong CbN is leftmost reduction and its key property is precisely that it is normalizing. A further contribution of the paper is an *untyped normalization theorem* for the external strategy in the Strong VSC, obtained as an easy corollary of the study via multi types. Such a result gives to the external strategy the same solid status of the leftmost strategy in CbN, and completes the picture.

No Tight Bounds. Multi types can be used to extract *tight bounds* on the length of evaluations and the size of normal forms. Here, we only study termination, not tight bounds, even if in the technical report [9] we also developed the enriched results with tight bounds. A first reason is that the characterization of external termination and the untyped normalization theorem we focus on here do not need the bounds. A second reason is that the enriched results are considerably more technical, while here we aim at a slightly weaker but more accessible treatment.

Proofs. Omitted proofs are in [10], the long version of this paper.

2 Technical Preliminaries

Basic Rewriting Notions. For a relation R on a set of terms, R^* is its reflexive-transitive closure. Given a relation \rightarrow_r , an *r-evaluation* (or simply evaluation if unambiguous) d is a finite sequence of terms $(t_i)_{0 \leq i \leq n}$ (for some $n \geq 0$) such that $t_i \rightarrow_r t_{i+1}$ for all $1 \leq i < n$, and we write $d: t \rightarrow_r^* u$ if $t_0 = t$ and $t_n = u$. The *length* n of d is noted $|d|$, and $|d|_a$ is the number of *a-steps* (i.e. the number of $t_i \rightarrow_a t_{i+1}$ for some $1 \leq i \leq n$) in d , for a given subrelation \rightarrow_a of \rightarrow_r .

A term t is *r-normal* if there is no u such that $t \rightarrow_r u$. An evaluation $d: t \rightarrow_r^* u$ is *r-normalizing* if u is r-normal. A term t is *weakly r-normalizing* if there is a r-normalizing evaluation $d: t \rightarrow_r^* u$; and t is *strongly r-normalizing* if there is no infinite sequence $(t_i)_{i \in \mathbb{N}}$ such that $t_0 = t$ and $t_i \rightarrow_r t_{i+1}$ for all $i \in \mathbb{N}$. Clearly, strong r-normalization implies weak r-normalization.

The Diamond Property. Following Dal Lago and Martini [22], we say that a relation \rightarrow_r is *diamond* if $u_1 \leftarrow_r t \rightarrow_r u_2$ and $u_1 \neq u_2$ imply $u_1 \rightarrow_r s \leftarrow_r u_2$ for some s . Terminology in the literature is inconsistent: Terese [44, Exercise 1.3.18] dubs this property CR^1 , and defines the diamond more restrictively, without requiring $u_1 \neq u_2$ in the hypothesis: u_1 and u_2 have to join even if $u_1 = u_2$.

Dal Lago and Martini show that if \rightarrow_r is diamond then:

1. \rightarrow_r is confluent, that is, $u_1 \leftarrow_r^* t \rightarrow_r^* u_2$ implies $u_1 \rightarrow_r^* s \leftarrow_r^* u_2$ for some s ;
2. all r-evaluations with the same start and r-normal end terms have the same length (i.e. if $d: t \rightarrow_r^* u$ and $d': t \rightarrow_r^* u$ with u r-normal, then $|d| = |d'|$);
3. t is weakly r-normalizing if and only if it is strongly r-normalizing.

Properties 2 and 3 are called *length invariance* and *uniform normalization*, respectively. Basically, the diamond captures a more liberal form of determinism.

Contextual Equivalence. The standard of reference for program equivalences is contextual equivalence, that can be defined abstractly as follows.

Definition 1 (Contextual Preorder and Equivalence). *Given a language of terms \mathcal{T} with its associated notion of contexts C and an operational semantics given as a rewriting relation \rightarrow , we define the associated contextual preorder \lesssim_C and contextual equivalence \simeq_C as follows:*

- $t \lesssim_C t'$ if $C\langle t \rangle$ weakly \rightarrow -normalizing implies $C\langle t' \rangle$ weakly \rightarrow -normalizing for all contexts C such that $C\langle t \rangle$ and $C\langle t' \rangle$ are closed terms.
- \simeq_C is the equivalence relation induced by $\lesssim_C: t \simeq_C t' \iff t \lesssim_C t' \text{ and } t' \lesssim_C t$.

TERMS	$t, u, s, q ::= x \mid \lambda x.t \mid tu$				
VALUES	$v, v' ::= \lambda x.t$				
EVALUATION CONTEXTS	$E ::= \langle \cdot \rangle \mid tE \mid Et$				
<table style="border: none; width: 100%; margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 20px;">RULE AT TOP LEVEL</td> <td>CONTEXTUAL CLOSURE</td> </tr> <tr> <td>$(\lambda x.t)v \mapsto_{\beta_v} t\{x \leftarrow v\}$</td> <td>$E\langle t \rangle \rightarrow_{\beta_v} E\langle u \rangle$ if $t \mapsto_{\beta_v} u$</td> </tr> </table>		RULE AT TOP LEVEL	CONTEXTUAL CLOSURE	$(\lambda x.t)v \mapsto_{\beta_v} t\{x \leftarrow v\}$	$E\langle t \rangle \rightarrow_{\beta_v} E\langle u \rangle$ if $t \mapsto_{\beta_v} u$
RULE AT TOP LEVEL	CONTEXTUAL CLOSURE				
$(\lambda x.t)v \mapsto_{\beta_v} t\{x \leftarrow v\}$	$E\langle t \rangle \rightarrow_{\beta_v} E\langle u \rangle$ if $t \mapsto_{\beta_v} u$				

Fig. 1: Our presentation of Plotkin’s calculus λ_v .

3 Call-by-Value and Call-by-Fireball

The call-by-value λ -calculus λ_v was introduced by Plotkin [40] in 1975 as the restriction of the λ -calculus where β -redexes can be fired only when their argument is a *value*, and values are defined as variables and abstractions.

Our Presentation of CbV. In Fig. 1, we present λ_v adopting three specific choices, departing from Plotkin’s presentation [40] in inessential details. Firstly, Plotkin also considers constants in the term syntax, with their own reduction rules, which are left unspecified. For simplicity, in our presentation there are no constants.

Secondly, for Plotkin values are variables and abstractions, while here *values are only abstractions*, as it is the case in the papers about Strong CbV motivating our study [12,14,15]. As stressed by Accattoli and Guerrieri [8], removing variables from values provides a better inductive description of normal forms in the open/strong case, without affecting properties such as termination or confluence. For further quantitative benefits, see Accattoli and Sacerdoti Coen [13]. In our paper, variables are not values, but all our results could be restated by considering them as values.

Thirdly, Plotkin defines both a multi-step non-deterministic (and confluent) evaluation relation reducing redexes *everywhere* in the term, and a single-step deterministic reduction (proceeding left-to-right) that is *weak*, that is, it does not reduce under abstractions. We instead adopt the somewhat halfway approach by Dal Lago and Martini [22]: our \rightarrow_{β_v} is single-step, weak but *non-deterministic*. The idea is that it can evaluate the left and right sub-term of an application in any order (that is, the left sub-term is not forced to be evaluated first as Plotkin does), because (in the weak case) the obtained reduction relation has the *diamond property* (definition in Section 2), a relaxed form of determinism. Thus, the obtained notion of reduction slightly generalizes Plotkin’s single-step reduction without changing whether a term terminates or not. The non-deterministic version is obtained via the notion of *evaluation context* E defined in Fig. 1.

Problem with Open Terms. It is well known that Plotkin’s framework works well only as long as terms are closed. The problem with open terms is that there are *false normal forms* such as $\Omega_l := (\lambda x.\delta)(yy)\delta$ (where $\delta := \lambda x.xx$ is the duplicator) which are β_v -normal, because yy is not a value (and cannot become one), but are *semantically divergent*. Such a divergence can be formalized in various ways, perhaps the simplest of which is that Ω_l is contextually equivalent

TERMS, VALUES, EVALUATION CTXS, and \rightarrow_{β_v} FIREBALLS INERT TERMS	As for Plotkin's calculus $f, f', f'' ::= v \mid i$ $i, i', i'' ::= x f_1 \dots f_n \quad n \geq 0$
RULE AT TOP LEVEL $(\lambda x.t)i \mapsto_{\beta_i} t\{x \leftarrow i\}$	CONTEXTUAL CLOSURE $E\langle t \rangle \rightarrow_{\beta_i} E\langle u \rangle \quad \text{if } t \mapsto_{\beta_i} u$

Fig. 2: The fireball calculus λ_{fire} .

(definition in Section 2) to the diverging term $\Omega := \delta\delta$. This problem was first pointed out by Paolini and Ronchi Della Rocca [39,38,41]. It is discussed at length by Accattoli and Guerrieri [5], who analyze various ways of extending Plotkin's calculus to solve the issue, that is, as to make false normal forms such as Ω_l diverge without simply switching to CbN (which, intuitively, amounts to diverge on $(\lambda x.y)\Omega$ as done by CbV but not by CbN), and show their equivalence with respect to termination, referring to them collectively as *Open CbV*.

The Fireball Calculus. The simplest presentation of Open CbV is probably the fireball calculus, defined in Fig. 2. The idea is that the values of the CbV λ -calculus are generalised to *fireballs* (a pun on *fire-able* terms), by adding *inert terms*, which contain in particular variables. Actually fireballs and inert terms are defined by mutual induction (in Fig. 2). For instance, $\lambda x.t$ is a fireball as a value, while x , $y(\lambda x.x)$, xy and $z(\lambda x.x)(zz)(\lambda y.t)$ are fireballs as inert terms.

The main feature of inert terms is that they are open, normal, and that when plugged in a context they cannot create a redex, hence the name. Essentially, they are the *neutral terms* of Open CbV.

Dynamically, β -redexes can also be fired if their argument is an inert term, via \rightarrow_{β_i} . Evaluation is weak, as evaluation contexts do not go under abstractions. Note that Ω_l given above now diverges: $\Omega_l = (\lambda x.\delta)(yy)\delta \rightarrow_{\beta_i} \delta\delta \rightarrow_{\beta_v} \dots$

Two relevant properties of the fireball calculus are that its reduction is non-deterministic but diamond and that a term is a normal form if and only if it is a fireball, a property called *harmony* by Accattoli and Guerrieri [5].

Issues with the Fireball Calculus. In later works [6,8], Accattoli and Guerrieri show that, despite the termination equivalence of the various formalisms for Open CbV, they can behave quite differently with respect to semantical notions. In particular, they show that the fireball calculus does not behave well with respect to Ehrhard's multi types [25], which are a notion of type inducing a paradigmatic denotational model of the λ -calculus, the *relational model*, linked to linear logic. Technically, they show that multi types do not satisfy subject reduction with respect to the fireball calculus. The issue can be illustrated without multi types, and amounts to the fact that the fireball calculus allows one to erase and duplicate inert terms. The erasure is particularly problematic. We reformulate here the problem in terms of contextual equivalence, to stress that the issue concerns the fireball calculus independently of multi types.

An expected property of any calculus is what we call here *contextual stability*, that is, the fact that its operational semantics \rightarrow is included in its contextual equivalence \simeq_c —in symbols: if $t \rightarrow^* u$ then $t \simeq_c u$. In the fireball calculus, contextual stability fails. Consider $t := (\lambda x.l)(yy)$, where $l := \lambda z.z$ is the identity combinator, and note that $t \rightarrow_{\beta_i} l$ because yy is an inert term. Now, consider the closing context $C := (\lambda y.(\cdot))\delta$ and note that $C\langle l \rangle = (\lambda y.l)\delta \rightarrow_{\beta_v} l$, while

$$C\langle t \rangle = (\lambda y.((\lambda x.l)(yy)))\delta \rightarrow_{\beta_v} (\lambda x.l)(\delta\delta) \rightarrow_{\beta_v} (\lambda x.l)(\delta\delta) \rightarrow_{\beta_v} \dots$$

That is, $t \rightarrow_{\beta_i} l$ but $t \not\simeq_c l$ because $C\langle t \rangle$ diverges while $C\langle l \rangle$ terminates.

To overcome false normal forms such as Ω_l , one *has to* work around redexes having inert terms has arguments. But substituting inert terms (thus sometimes erasing them) as done by the fireball calculus is a *brute force* solution. At the open level, it does not alter termination but it deteriorates semantics properties of the calculus such as contextual stability or the relationship with multi types. The alternative presentation of Open CbV given by the value substitution calculus [11], discussed in the next section, does not suffer of these shortcomings.

Issues with the Strong Fireball Calculus. The strong evaluation strategy by Biernacka et al. [14,15] is a deterministic version of the fireball calculus (namely proceeding right-to-left) extended to evaluate under abstractions. We omit the actual definition of its extension under abstractions because it is non-trivial and the issue we want to point out can be explained without detailing them.

At the open level, the issue with the fireball calculus is about the semantics, but not about termination. The semantical issue of the open level, however, induces a termination issue at the strong level. Indeed, the strong fireball calculus suffers of a phenomenon similar to the one of false normal forms, even if no redexes are stuck. Consider the term $u := (\lambda x.l)(y(\lambda z.\Omega))$. In the fireball calculus, one has $u \rightarrow_{\beta_i} l$ because $y(\lambda z.\Omega)$ is an inert term. That is, u terminates on a strong normal form, namely l . Instead, u is *semantically divergent* at the strong level, as it was the case for Ω_l in Plotkin’s calculus. Intuitively, the inert sub-term $y(\lambda z.\Omega)$ should be somehow kept, instead of being erased, and evaluated strongly, which would lead to divergence because of the Ω under abstraction.

The non-trivial point is how to bring evidence that this is what should happen in a good definition of strong CbV evaluation. One way to see that something is wrong with the strong fireball calculus is to observe that the step $u \rightarrow_{\beta_i} l$ provides another breaking of contextual stability. This is detailed in the Appendix of [10], because although interesting it does not involve strong evaluation.

Better evidence is developed along the paper. We adopt the value substitution calculus (VSC) and the external strategy by Accattoli et al. [2] for which t is divergent. Then, we show that, when refining Ehrhard’s CbV multi types [25] by adding the machinery for characterizing termination of strong evaluation, one obtains that t above is not typable. That is, t is also semantically diverging.

4 Value Substitution Calculus

Here we present the *value substitution calculus* (VSC for short) introduced by Accattoli and Paolini [11], and recall some properties. The operational semantics shall be different from the fireball calculus of the previous section, but we shall nonetheless exploit the concept of fireball to describe its normal forms.

Terms. The VSC is a CbV λ -calculus extended with let-expressions, similarly to Moggi's CbV calculus [36,37]. We do however write a let-expression $\text{let } x = u \text{ in } t$ as a more compact *explicit substitution* $t[x \leftarrow u]$ (ES for short), which binds x in t . Moreover, our let/ES does not fix an order of evaluation between t and u , in contrast to many papers in the literature (*e.g.* Sabry and Wadler [43] or Levy et al. [33]) where u is evaluated first. The grammars follow:

$$\text{VALUES } v ::= \lambda x.t \qquad \text{TERMS } t, u, s ::= x \mid v \mid tu \mid t[x \leftarrow u]$$

The set of free variables of term t is denoted by $\text{fv}(t)$ and terms are identified up to α -renaming of bound variables. We use $t\{x \leftarrow u\}$ for the capture-avoiding substitution of u for each free occurrence of the variable x in t .

Contexts. All along the paper, we use (many notions of) *contexts*, *i.e.* terms with exactly one hole, noted $\langle \cdot \rangle$. For now, we need general contexts and the notion of *substitution contexts* L , which are simply lists of ES. The grammars are:

$$\begin{array}{ll} \text{(GENERAL) CONTEXTS} & C ::= \langle \cdot \rangle \mid Ct \mid tC \mid \lambda x.C \mid C[x \leftarrow t] \mid t[x \leftarrow C] \\ \text{SUBSTITUTION CONTEXTS} & L ::= \langle \cdot \rangle \mid L[x \leftarrow t] \end{array}$$

Plugging a term t in a context C is noted $C(t)$, possibly capturing variables, for instance $(\lambda x.\lambda y.\langle \cdot \rangle)\langle xy \rangle = \lambda x.\lambda y.xy$ (while $(\lambda x.\lambda y.z)\{z \leftarrow xy\} = \lambda x'.\lambda y'.xy$). An *answer* is a term of shape $L\langle v \rangle$.

Reduction Rules. The reduction rules of VSC are slightly unusual as they use *contexts* both to allow one to reduce redexes located in sub-terms, which is standard, *and* to define the redexes themselves, which is less standard—this kind of rule is called *at a distance*. The rewriting rules in fact mimic exactly cut-elimination on proof nets, via Girard's CbV translation $(A \Rightarrow B)^v = !(A^v \multimap B^v)$ [26] of intuitionistic logic into linear logic, see Accattoli [1].

There are two rewrite rules. Their root cases (that is, before context closure) follow (the terminology is inherited from linear logic):

$$\begin{array}{ll} \text{MULTIPLICATIVE ROOT RULE} & L\langle \lambda x.t \rangle u \mapsto_m L\langle t[x \leftarrow u] \rangle \\ \text{EXPONENTIAL ROOT RULE} & t[x \leftarrow L\langle v \rangle] \mapsto_e L\langle t\{x \leftarrow v\} \rangle \end{array}$$

Both root rules are *at a distance* in that they involve a substitution context L , and L does not capture the free variables of u in \mapsto_m (resp. of t in \mapsto_e). We shall consider two variants of the VSC, the open and the strong version. They differ only in the choice of evaluation contexts for the root rewrite rules.

The Open VSC. We first focus on the open fragment of the VSC, where rewriting is forbidden under abstraction and terms are *possibly* open (but not necessarily). This fragment has a nice inductive description of its normal forms, called *fireballs* and *inert terms* as they are the lifting to the VSC of the respective notions from the fireball calculus. Open contexts and rules are defined as follows.

$$\begin{array}{l} \text{OPEN CONTEXTS } O ::= \langle \cdot \rangle \mid Ot \mid tO \mid O[x \leftarrow t] \mid t[x \leftarrow O] \\ \text{OPEN REWRITE RULES: } \frac{t \mapsto_a t'}{O\langle t \rangle \rightarrow_{oa} O\langle t' \rangle} \quad (a \in \{\mathbf{m}, \mathbf{e}\}) \quad \text{OPEN REDUCTION: } \rightarrow_{\circ} := \rightarrow_{\text{om}} \cup \rightarrow_{\text{oe}} \end{array}$$

Proposition 1 (Properties of open reduction [8]).

1. Open reduction \rightarrow_{\circ} is diamond.
2. A term is \circ -normal if and only if it is a fireball, where fireballs (and inert terms) are defined by:

$$\text{INERT TERMS } i, i' ::= x \mid if \mid i[x \leftarrow i'] \quad \text{FIREBALLS } f ::= v \mid i \mid f[x \leftarrow i]$$

Plotkin vs VSC. The open fragment of the VSC is enough to discuss the relationship with Plotkin's CbV λ -calculus as defined in the previous section. Plotkin's calculus can be easily simulated in the VSC. Indeed, if $(\lambda x.t)v \mapsto_{\beta_v} t\{x \leftarrow v\}$ then $(\lambda x.t)v \mapsto_{\mathbf{m}} t[x \leftarrow v] \mapsto_{\mathbf{e}} t\{x \leftarrow v\}$.

There is no sensible way, instead, to simulate VSC into Plotkin's calculus. Indeed, VSC is a proper extension of Plotkin's: false normal forms of Plotkin's calculus such as $\Omega_l = (\lambda x.\delta)(yy)\delta$ and $\Omega_r := \delta((\lambda x.\delta)(yy))$ are divergent in VSC:

$$\begin{array}{l} \Omega_l \rightarrow_{\text{om}} \delta[x \leftarrow yy]\delta \rightarrow_{\text{om}} (xx)[x \leftarrow \delta][x \leftarrow yy] \rightarrow_{\text{oe}} (\delta\delta)[x \leftarrow yy] \rightarrow_{\text{om}} \dots \\ \Omega_r \rightarrow_{\text{om}} \delta\delta[x \leftarrow yy] \rightarrow_{\text{om}} (xx)[x \leftarrow \delta][x \leftarrow yy] \rightarrow_{\text{oe}} (\delta\delta)[x \leftarrow yy] \rightarrow_{\text{om}} \dots \end{array}$$

Note that divergence of Ω_l crucially uses distance on $\rightarrow_{\mathbf{m}}$ (in the second step), while divergence of Ω_r crucially uses distance on $\rightarrow_{\mathbf{e}}$ (in the third step).

VSC and Contextual Equivalence. Pleasantly, Plotkin's calculus and the VSC induce the same notion of contextual equivalence on λ -terms without ES, since contextual equivalence is defined with respect to contexts that *close* terms, see Accattoli and Guerrieri [8]. Moreover, the Open VSC is *contextually stable*.

Proposition 2 (Contextual stability [3]). *The Open VSC is contextually stable, that is, if $t \rightarrow_{\circ}^* u$ then $t \simeq_{\mathbf{c}} u$.*

The Strong VSC. The Strong VSC is obtained by allowing rewriting rules everywhere, including under abstractions, via a closure by general contexts.

$$\begin{array}{l} \text{STRONG REWRITE RULES: } \frac{t \mapsto_a t'}{C\langle t \rangle \rightarrow_a C\langle t' \rangle} \quad (a \in \{\mathbf{m}, \mathbf{e}\}) \quad \text{STRONG REDUCTION: } \rightarrow_{\text{vsc}} := \rightarrow_{\mathbf{m}} \cup \rightarrow_{\mathbf{e}} \end{array}$$

Unlike the previous cases, \rightarrow_{vsc} is not diamond: consider all the vsc-evaluations of $(xx)[x \leftarrow \lambda y.\mathbb{I}]$, with $\mathbb{I} := \lambda z.z$.

Proposition 3 (Properties of strong reduction [11,8]).

1. The reduction \rightarrow_{vsc} is confluent.
2. A term is vsc-normal if and only if it is a strong fireball, where strong fireballs (and strong inert terms, strong values) are:

$$\begin{array}{l} \text{STRONG INERT TERMS } i_s ::= x \mid i_s f_s \mid i_s[x \leftarrow i'_s] \\ \text{STRONG FIREBALLS } f_s ::= i_s \mid v_s \mid f_s[x \leftarrow i_s] \end{array} \quad \begin{array}{l} \text{STRONG VALUES } v_s ::= \lambda x. f_s \end{array}$$

The notions of strong inert terms and strong fireballs are a generalization of inert terms and fireballs, respectively, by simply iterating the construction under all abstractions. Note that they are similar to normal forms of the (CbN) λ -calculus, but they can have ESs containing strong inert terms.

5 The External Strategy

In this section, we define Accattoli et al.'s (strong) *external* strategy [2], that shall be studied via multi types in Sect. 8. Its role is analogous to the leftmost-outermost strategy of the λ -calculus. A notable difference, however, is that the external strategy is itself non-deterministic, but in a harmless way, because it is diamond. The idea is the same used for Plotkin's calculus, that is, allowing one to reduce sub-terms of applications (and ES) in any order. In a strong setting, however, it is a bit trickier to enforce it.

We need a few notions. Firstly, *rigid terms*, *i.e.* the variation over inert terms where the arguments of the head variable can be whatever term:

$$\text{RIGID TERMS } r, r' ::= x \mid rt \mid r[x \leftarrow r']$$

Every (strong) inert term is a rigid term, but the converse does not hold—consider $y(\delta!)$, which is rigid but not inert.

Secondly, we need evaluation contexts for the external strategy \rightarrow_x , which is defined on top of open evaluation. The base case is given by the open rewriting rules (themselves defined via a closure by open contexts, see Section 4), which are then closed by *external contexts*, defined mutually with *rigid contexts*:

$$\begin{array}{l} \text{EXTERNAL CONTEXTS } X ::= \langle \cdot \rangle \mid \lambda x. X \mid t[x \leftarrow R] \mid X[x \leftarrow r] \mid R \\ \text{RIGID CONTEXTS } R ::= rX \mid Rt \mid R[x \leftarrow r] \mid r[x \leftarrow R] \end{array}$$

$$\begin{array}{l} \text{EXTERNAL REWRITE RULES: } \frac{t \rightarrow_{\text{oa}} t'}{(a \in \{\text{m}, \text{e}\}) X \langle t \rangle \rightarrow_{\text{xa}} X \langle t' \rangle} \quad \text{EXTERNAL REDUCTION: } \\ \rightarrow_x := \rightarrow_{\text{xm}} \cup \rightarrow_{\text{xe}} \end{array}$$

Clearly, $\rightarrow_x \not\subseteq \rightarrow_{\text{vsc}}$. The strategy diverges on $y(\lambda z. \Omega)$ (as $y\lambda z. \langle \cdot \rangle$ is a rigid—thus external—context) and normalizes the potentially diverging term $(\lambda x. y)(\lambda z. \Omega) \rightarrow_x^* y$, because values can be erased even if they diverge under abstraction.

Key example: the external strategy diverges on the term $t = (\lambda x. !)(y(\lambda z. \Omega))$ of Sect. 3 on which the strong fireball calculus terminates, showing that *the two strong settings have different notions of termination*. Indeed, $t \rightarrow_{\text{xm}} ! [x \leftarrow y(\lambda z. \Omega)]$

and then the external strategy diverges because $y(\lambda z.\Omega)$ cannot be erased and Ω occurs in the external evaluation context $\mathbb{I}[x \leftarrow y(\lambda z.\langle \cdot \rangle)]$.

The grammars of external and rigid contexts allow evaluation to enter only inside non-applied abstractions, *e.g.* $(\lambda x.(\mathbb{I}))v \not\rightarrow_x (\lambda x.(y[y \leftarrow \mathbb{I}]))v$. This is a sort of outside-in order which is neither left-to-right nor right-to-left—we have both $(\mathbb{I})(\mathbb{I}) \rightarrow_{xm} (y[y \leftarrow \mathbb{I}])(\mathbb{I})$ and $(\mathbb{I})(\mathbb{I}) \rightarrow_{xm} (\mathbb{I})(y[y \leftarrow \mathbb{I}])$ —since open contexts do not impose an order on applications. As just showed, the strategy is non-deterministic: another example is given by $t = x(\lambda y.(\mathbb{I}))[x \leftarrow x(\mathbb{I})] \rightarrow_{xm} x(\lambda y.z[z \leftarrow \mathbb{I}])[x \leftarrow x(\mathbb{I})]$, and $t \rightarrow_{xm} x(\lambda y.(\mathbb{I}))[x \leftarrow x(z[z \leftarrow \mathbb{I}])]$. Such a behavior however is only a relaxed form of determinism, as it satisfies the *diamond property*.

Proposition 4 (Properties of external reduction \rightarrow_x [2]).

1. *External reduction \rightarrow_x is diamond.*
2. *Fullness: let t be a VSC term, t is x -normal if and only if t is vsc-normal.*

6 Multi Types by Value

We present here the system of CbV *multi types* that we shall use to characterize the termination of the external strategy in Sect. 8. The system was introduced by Ehrhard [25] for Plotkin’s CbV λ -calculus, as the CbV version of de Carvalho’s multi types system for CbN [18,19]. Both systems can be seen as presentations of the relational semantics of linear logic restricted to the CbV/CbN interpretation of the λ -calculus. The CbV multi type system is also used in [23,17,28,6,8,3].

Multi Types. There are two layers of types, *linear* and *multi types*:

$$\begin{array}{ll} \text{LINEAR TYPES} & \mathbf{A}, \mathbf{B} ::= \mathbf{G} \mid \mathbf{M} \multimap \mathbf{N} \\ \text{MULTI TYPES} & \mathbf{M}, \mathbf{N} ::= [\mathbf{A}_1, \dots, \mathbf{A}_n] \quad n \in \mathbb{N} \end{array}$$

where \mathbf{G} is an unspecified ground type and $[\mathbf{A}_1, \dots, \mathbf{A}_n]$ is our notation for finite multisets. The *empty* multi type $[\]$ obtained by taking $n = 0$ is also denoted by $\mathbf{0}$. When CbV multi types are used to study weak evaluation, they are usually presented without the ground type \mathbf{G} , as one can use $\mathbf{0}$ as base case for types. For studying strong evaluation, however, \mathbf{G} is mandatory, as we shall see.

A multi type $[\mathbf{A}_1, \dots, \mathbf{A}_n]$ has to be intended as a conjunction $\mathbf{A}_1 \wedge \dots \wedge \mathbf{A}_n$ of linear types $\mathbf{A}_1, \dots, \mathbf{A}_n$, for a commutative, associative, non-idempotent conjunction \wedge (morally a tensor \otimes), of neutral element $\mathbf{0}$. Note however that $[\mathbf{A}] \neq \mathbf{A}$.

The intuition is that a linear type corresponds to a single use of a term t , and that t is typed with a multiset \mathbf{M} of n linear types if it is going to be used (at most) n times. The meaning of *using a term* (once) is not easy to define precisely. Roughly, it means that if t is part of a larger term u , then a copy of t shall end up in evaluation position during the evaluation of u . More precisely, the copy shall end up in evaluation position where it is applied to some terms.

The derivation rules for the multi types system are in Figure 3 (explanation follows). The rules are the same as in Ehrhard [25], up to their extension to ESs.

Judgments have shape $\Gamma \vdash t : \mathbf{M}$ or $\Gamma \vdash t : \mathbf{A}$ where t is a term, \mathbf{M} is a multi type, \mathbf{A} is a linear type, and Γ is a *type context*, that is, a total function from variables to multi types such that $\text{dom}(\Gamma) := \{x \mid \Gamma(x) \neq \mathbf{0}\}$ is finite.

$$\begin{array}{c}
\frac{}{x:[A] \vdash x:A} \text{ax} \qquad \frac{\Gamma \vdash t:[M \multimap N] \quad \Delta \vdash u:M}{\Gamma \uplus \Delta \vdash tu:N} \textcircled{\text{a}} \qquad \frac{\Gamma, x:M \vdash t:N}{\Gamma \vdash \lambda x.t:M \multimap N} \lambda \\
\frac{\Gamma, x:M \vdash t:N \quad \Delta \vdash u:M}{\Gamma \uplus \Delta \vdash t[x \leftarrow u]:N} \text{es} \qquad \frac{[\Gamma_i \vdash v_t:A_i]_{i \in I}}{\uplus_{i \in I} \Gamma_i \vdash v_t:[A_i]_{i \in I}} \text{many}
\end{array}$$

Fig. 3: Call-by-value multi type system. In rule *many*, v_t is a *theoretical value*, i.e. a variable or an abstraction, and I is a finite set.

Technicalities about Types. The type context Γ is *empty* if $\text{dom}(\Gamma) = \emptyset$. *Multiset sum* \uplus is extended to type contexts point-wise, i.e. $(\Gamma \uplus \Delta)(x) := \Gamma(x) \uplus \Delta(x)$ for each variable x . This notion is extended to a finite family of type contexts as expected, in particular $\uplus_{i \in J} \Gamma_i$ is the empty type context if $J = \emptyset$. A type context Γ is denoted by $x_1:M_1, \dots, x_n:M_n$ (for some $n \in \mathbb{N}$) if $\text{dom}(\Gamma) \subseteq \{x_1, \dots, x_n\}$ and $\Gamma(x_i) = M_i$ for all $1 \leq i \leq n$. Given two type contexts Γ and Δ such that $\text{dom}(\Gamma) \cap \text{dom}(\Delta) = \emptyset$, the type context Γ, Δ is defined by $(\Gamma, \Delta)(x) := \Gamma(x)$ if $x \in \text{dom}(\Gamma)$, $(\Gamma, \Delta)(x) := \Delta(x)$ if $x \in \text{dom}(\Delta)$, and $(\Gamma, \Delta)(x) := \mathbf{0}$ otherwise. Note that $\Gamma, x:\mathbf{0} = \Gamma$, where we implicitly assume $x \notin \text{dom}(\Gamma)$.

We write $\Phi \triangleright \Gamma \vdash t:M$ if Φ is a (*type*) *derivation* (i.e. a tree constructed using the rules in Figure 3) with conclusion the multi judgment $\Gamma \vdash t:M$. In particular, we write $\Phi \triangleright \vdash t:M$ when Γ is empty. We write $\Phi \triangleright t$ if $\Phi \triangleright \Gamma \vdash t:M$ for some type context Γ and some multi type M .

We need a notion of size of type derivations, which shall be used as the termination measure for typable terms.

Definition 2 (Derivation size). *Let Φ be a type derivation. The size $|\Phi|$ of Φ is the number of rule occurrences in Φ except for rule *many*.*

Multisets and Rule many. Rule *many* plays a crucial role, as it is the only rule introducing multisets on the right-hand side of judgments: it takes as premises a finite multiset of derivations of linear types for a term v_t , and glues them together giving a judgment with the finite multiset of linear types to v_t . The term v_t is a *theoretical value* v_t , that is, a variable or an abstraction—the terminology is taken from Accattoli and Sacerdoti Coen [13]. Rule *many* is the multi types analogous of the promotion rule of linear logic, which, in the CbV representation of the λ -calculus, is indeed used for typing abstractions and variables. Note that in particular all abstractions are typable with $\mathbf{0}$ via a *many* rule with no premises.

Subject Reduction and Expansion. The first properties of the type system that we show are subject reduction and expansion, which hold for *every* VSC step, not only external ones. They rely on a substitution lemma (and its inverse for subject expansion) in the Appendix of [10].

Proposition 5 (Qualitative subjects). *Let $t \rightarrow_{\text{vsc}} t'$.*

1. Reduction: *if $\Phi \triangleright \Gamma \vdash t:M$ then there is $\Phi' \triangleright \Gamma \vdash t':M$ such that $|\Phi| \geq |\Phi'|$.*

2. Expansion: if $\Phi' \triangleright \Gamma \vdash t' : \mathbf{M}$ then there is a derivation $\Phi \triangleright \Gamma \vdash t : \mathbf{M}$.

Note that subject reduction (Prop. 5.1) says also that the derivation size *cannot increase* after a reduction step. It does not say that it *decreases* at every step because, for instance, if $\lambda x.t \rightarrow_{\text{vsc}} \lambda x.t'$ and $\lambda x.t$ is typed using a empty many rule (*i.e.* with 0 premises), which is a derivation of size 0, then also $\lambda x.t'$ is typed using a empty many rule, of size 0. Hence, not all typable terms terminate for strong/external evaluation, as for instance $\lambda x.\Omega$ is typable (with $\mathbf{0}$).

There are two ways to strengthen subject reduction without changing the type system and recover termination: restricting either the reductions to not take place under abstraction, which is what we shall do in the next section for the open case, or the kind of types taken into account (roughly, as to limit the use of $\mathbf{0}$), which is what shall guarantee termination for the external strategy.

7 Multi Types for Open CbV

Here we recall the qualitative part of the relationship between CbV multi types and Open CbV studied by Accattoli and Guerrieri in [8], where they develop also a quantitative study not used here. The reason to recall their result is twofold. Firstly, the external case relies on the open one. Secondly, the open case provides the blueprint for the strong case, allowing us to stress similarities and differences.

The result is that the open evaluation \rightarrow_{\circ} of t terminates if and only if t is typable. Since \rightarrow_{\circ} does not reduce under abstractions, every abstraction is \circ -normal and indeed typable, for instance with $\mathbf{0}$. As an example, note that $\lambda x.\Omega$ is typable with $\mathbf{0}$ (rule many with 0 premises), though Ω is not.

Correctness. Open correctness establishes that all typable terms are \circ -normalizing and it is proved by showing that the size of type derivation decreases with every \circ -step. Open correctness is proved following a standard scheme, namely proving a quantitative version of open subject reduction, stating that every \rightarrow_{\circ} step preserves types and decreases the general size of a derivation.

Proposition 6 (Open quantitative subject reduction). *Let $\Phi \triangleright \Gamma \vdash t : \mathbf{M}$ be a derivation. If $t \rightarrow_{\circ} t'$ then there is $\Phi' \triangleright \Gamma \vdash t' : \mathbf{M}$ with $|\Phi| > |\Phi'|$.*

The size of derivations decreases after any \rightarrow_{\circ} step, thus proving \rightarrow_{\circ} -termination for typable terms. Clearly, the size provides a bound to the number of steps.

Theorem 1 (Open correctness). *Let $\Phi \triangleright t$ be a derivation. Then there is a \circ -normalizing evaluation $d : t \rightarrow_{\circ}^* f$ with $|d| \leq |\Phi|$.*

Completeness. Open completeness establishes that every \circ -normalizing term is typable. Again, the proof technique is standard: a lemma states that every \circ -normal form is typable, and subject expansion (Prop. 5.2) allows us to pull back typability along \rightarrow_{\circ} steps. The lemma about open normal forms says that they are all typable with $\mathbf{0}$ and relies on a stronger statement about inert terms: they can be assigned whatever multi type \mathbf{M} , by tuning the type context Γ accordingly.

RIGHT MULTI TYPE $M^R ::= [A_1^R, \dots, A_n^R] \quad n \geq 1$ RIGHT LINEAR TYPE $A^R ::= G \mid M^L \multimap M^R$
LEFT MULTI TYPE $M^L ::= [A_1^L, \dots, A_n^L] \quad n \geq 0$ LEFT LINEAR TYPE $A^L ::= G \mid M^R \multimap M^L$

Fig. 4: Right and left (shrinking) types.

Lemma 1 (Typability of open normal forms).

1. Inert: for every inert term i and multi type M , there exists a type context Γ and a derivation $\Phi \triangleright \Gamma \vdash i : M$.
2. Fireball: for every fireball f there exists a type context Γ and a derivation $\Phi \triangleright \Gamma \vdash f : \mathbf{0}$.

Theorem 2 (Open completeness). *Let $d : t \rightarrow_o^* f$ be an \mathbf{o} -normalizing evaluation. Then there is a derivation $\Phi \triangleright \Gamma \vdash t : \mathbf{0}$.*

8 Shrinking Multi Types for the External Strategy

In this section, we restrict the set of judgments as to characterize the typable terms that terminate with respect to the external strategy. The restriction is obtained by adapting to CbV the *shrinking technique* for CbN multi types in Accattoli et al. [4]. At the end of the section, we also obtain the untyped normalization theorem for the external strategy.

The definition of shrinking judgments is standard and not due to [4], see for instance Krivine’s book [32], but the proof technique that we shall use is due to [4] and it is different from others in the literature [32,19,31,16]. Its key ingredient is the isolation of a key property of rigid terms (Lemma 2 below).

The Need for Shrinking. As already pointed out at the end of Sect. 6, some terms that diverge with strong evaluation are typable. We have that Ω itself is not typable, but $\lambda y.\Omega$ is typable with $\mathbf{0}$ (via a many rule with 0 premises). It might seem that the problem is being typable with $\mathbf{0}$, but also $x(\lambda y.\Omega)$ is externally divergent and can be typed by assigning $[\mathbf{0} \multimap M]$ to x (any M works). In this case the problem is that, since $\mathbf{0}$ is on the left of \multimap , the argument is meant to be erased, but x cannot actually erase it. This is a problem typical of strong settings, as it occurs also in Strong CbN. The solution is to restrict to type derivations satisfying a predicate that forbids types where $\mathbf{0}$ plays these dangerous tricks; in particular a ground type $G \neq \mathbf{0}$ is needed. This is unavoidable and standard, see [32,4]. Following [4], the predicate is here called *shrinkingness* because it ensures that the size of type derivations shrinks at each \rightarrow_x step (see Prop. 7 below).

Defining Shrinking. The definition of shrinking forbids the empty multiset $\mathbf{0}$ on the left of some type arrows \multimap . We actually need two notions of shrinking types, *left* and *right*. Intuitively, it is because the typing rule λ shifts a type from the left-hand side of a judgment to the left of \multimap on the right-hand side of a judgment. Formally, *left* and *right shrinking* (multi or linear) *types* are defined

in Figure 4 (we omit *shrinking* when referring to left or right types, for brevity), by mutual induction. The key point is that right multi types *cannot be empty* (note $n \geq 1$), thus $\mathbf{0}$ is forbidden on the left of top \multimap for left linear types.

The notions extend to type contexts and to derivations as follows:

- A type context $x_1:M_1, \dots, x_n:M_n$ is *left* if each M_i is left;
- A derivation $\Phi \triangleright \Gamma \vdash t:M$ is *shrinking* if Γ is left and M is right.

Examples: $[\mathbf{G}]$ is both left and right (this fact shall play a role below), while $\mathbf{0}$ is left but not right, and $[\mathbf{0} \multimap [\mathbf{G}]]$ is right but not left.

Key Property of Left Shrinking. Shrinkingness is a predicate of derivations depending only on their *final judgment*. For proving properties of shrinking derivations, we have to analyze how shrinking propagates to sub-derivations, to apply the *i.h.* in proofs. The following lemma is specific to *left* shrinkingness, on which the propagation of shrinkingness then builds. It says that for specific terms—typable *rigid* terms—left shrinkingness spreads from the type context to the right-hand multi type in a judgment. It is the key property of the proof technique.

Lemma 2 (Spreading of left shrinkingness on judgments). *Let $\Phi \triangleright \Gamma \vdash r:M$ be a derivation and r be a rigid term. If Γ is left then M is left.*

Correctness. Shrinking correctness establishes that all typable terms with a shrinking derivation are externally normalizing. We follow the same pattern as for the open case, but the proof of subject reduction is trickier—this is the delicate point of the proof technique by Accattoli et al. [4]. It crucially uses the key property of left shrinking for rigid terms above (Lemma 2), and it also requires an auxiliary statement with a weaker hypothesis for the induction to go through.

Proposition 7 (Shrinking quantitative subject reduction for \rightarrow_x).

1. Auxiliary statement: *Let Γ be a left context. Suppose that $\Phi \triangleright \Gamma \vdash t:M$ and that if t is an answer then M is right. If $t \rightarrow_x t'$ then there is a derivation $\Phi' \triangleright \Gamma \vdash t':M$ with $|\Phi| > |\Phi'|$.*
2. Actual statement: *Let $\Phi \triangleright \Gamma \vdash t:M$ be a shrinking derivation. If $t \rightarrow_x t'$ then there is a derivation $\Phi' \triangleright \Gamma \vdash t':M$ with $|\Phi| > |\Phi'|$.*

Proof. Note that the auxiliary statement is stronger, because every shrinking derivation (defined as having Γ left and M right) satisfies it: if t is not an answer then M can be whatever, in particular it can be right. For the auxiliary statement, we give two cases, the one motivating the use of the auxiliary statement and one showing the use of the key property for rigid terms. The other cases are in the Appendix of [10]. The proof is by induction on the external context X such that $t = X\langle u \rangle \rightarrow_x X\langle u' \rangle = t'$ with $u \rightarrow_o u'$. The two cases:

- *Rigid context applied to term, i.e. $X = Rs$.* Then, $t = X\langle u \rangle = R\langle u \rangle s \rightarrow_x R\langle u' \rangle s = X\langle u' \rangle = t'$ with $u \rightarrow_o u'$. The derivation Φ has the following shape:

$$\Phi = \frac{\Psi \triangleright \Delta \vdash R\langle u \rangle : [\mathbf{N} \multimap \mathbf{M}] \quad \Theta \triangleright \Sigma \vdash s : \mathbf{N}}{\Delta \uplus \Sigma \vdash R\langle u \rangle s : \mathbf{M}} @$$

where $\Gamma = \Delta \uplus \Sigma$ is left by hypothesis, and then so is Δ . By *i.h.* (as $R\langle u \rangle$ is not an answer), there is a derivation $\Psi' \triangleright \Delta \vdash R\langle u' \rangle : [\mathbf{N} \multimap \mathbf{M}]$ with $|\Psi'| < |\Psi|$. We can then build the following derivation:

$$\Phi' = \frac{\Psi' \triangleright \Delta \vdash R\langle u' \rangle : [\mathbf{N} \multimap \mathbf{M}] \quad \Theta \triangleright \Sigma \vdash s : \mathbf{N}}{\Delta \uplus \Sigma \vdash R\langle u' \rangle s : \mathbf{M}} @$$

where $\Gamma = \Delta \uplus \Sigma$ and $|\Phi'| = |\Psi'| + |\Theta| + 1 < |\Psi| + |\Theta| + 1 = |\Phi|$. Note that in this case we have no hypothesis on \mathbf{N} , thus on $[\mathbf{N} \multimap \mathbf{M}]$, which is why we need a weaker statement in order to use the *i.h.*

- *Rigid term applied to external context, i.e. $X = rX'$.* Then, $t = X\langle u \rangle = rX'\langle u \rangle \rightarrow_x rX'\langle u' \rangle = X\langle u' \rangle = t'$ with $u \rightarrow_o u'$. The derivation Φ is:

$$\Phi = \frac{\Psi \triangleright \Delta \vdash r : [\mathbf{N} \multimap \mathbf{M}] \quad \Theta \triangleright \Sigma \vdash X'\langle u \rangle : \mathbf{N}}{\Delta \uplus \Sigma \vdash rX'\langle u \rangle : \mathbf{M}} @$$

where $\Gamma = \Delta \uplus \Sigma$ is left by hypothesis, and then so are Δ and Σ . According to spreading of left shrinkingness applied to Ψ (Lemma 2, which can be applied because r is a rigid term), $[\mathbf{N} \multimap \mathbf{M}]$ is a left multi type and hence \mathbf{N} is a right multi type. Thus, the *i.h.* applied to Θ gives a derivation $\Theta' \triangleright \Sigma \vdash X'\langle u' \rangle : \mathbf{N}$ with $|\Theta'| < |\Theta|$. We then build the following derivation:

$$\Phi' = \frac{\Psi \triangleright \Delta \vdash r : [\mathbf{N} \multimap \mathbf{M}] \quad \Theta' \triangleright \Sigma \vdash X'\langle u' \rangle : \mathbf{N}}{\Delta \uplus \Sigma \vdash rX'\langle u' \rangle : \mathbf{M}} @$$

where $\Gamma = \Delta \uplus \Sigma$ and $|\Phi'| = |\Psi| + |\Theta'| + 1 < |\Psi| + |\Theta| + 1 = |\Phi|$. \square

Theorem 3 (Shrinking correctness for \rightarrow_x). *Let $\Phi \triangleright t$ be a shrinking derivation. Then there is a x -normalizing evaluation $d : t \rightarrow_x^* f_s$ with $|d| \leq |\Phi|$.*

Shrinking correctness for \rightarrow_x shows that the term $t = (\lambda x.l)(y(\lambda z.\Omega))$ diverging for \rightarrow_x but normalizing for the strong fireball calculus is not typable, otherwise it would \rightarrow_x -terminate. That is, it shows that t is semantically diverging.

Completeness. Shrinking completeness is proven as in the open case, using a lemma about the shrinking typability of strong fireballs. Note that the lemma now has an existential quantification on the type \mathbf{M} of strong fireballs, while in the open case the type was simply $\mathbf{0}$; here $\mathbf{0}$ would not work, because it is not right. Note also the part about inert terms, stating that \mathbf{M} is *left*: it is not a mistake, it can be seen as an instance of the key properties of rigid terms (Lemma 2, inert terms are rigid), and it gives shrinking derivations when \mathbf{M} is instantiated with, say, $[\mathbf{G}]$, which is both left and right.

Lemma 3 (Shrinking typability of normal forms).

1. Inert: for every strong inert term i_s and left multi type \mathbf{M} , there exists a left type context Γ and a derivation $\Phi \triangleright \Gamma \vdash i_s : \mathbf{M}$.
2. Fireball: for every strong fireball f_s there is a shrinking derivation $\Phi \triangleright f_s$.

Theorem 4 (Shrinking completeness for \rightarrow_x). *Let $d : t \rightarrow_x^* f_s$ be a x -normalizing evaluation. Then there is a shrinking derivation $\Phi \triangleright t$.*

Untyped Normalization Theorem. By exploiting an elegant proof technique used by de Carvalho et al. [20] and Mazza et al. [35], we obtain an *untyped normalization* theorem for the external strategy of the VSC, as a corollary of our study of multi types. The key points are subject expansion (Prop. 5.2) for the *whole* reduction \rightarrow_{vsc} , instead that just for the external strategy, and the fact that shrinkingness is a predicate of derivations depending only on their conclusion.

Theorem 5 (Untyped normalization for \rightarrow_x). *If there is a vsc-normalizing evaluation $d: t \rightarrow_{\text{vsc}}^* f_s$, then $t \rightarrow_x^* f_s$.*

Proof. Shrinking typability of normal forms (Lemma 3) gives a shrinking derivation $\Phi \triangleright \Gamma \vdash f_s : \mathbb{M}$. Subject expansion (Prop. 5.2) iterated along $t \rightarrow_{\text{vsc}}^* f_s$ gives a shrinking derivation $\Psi \triangleright \Gamma \vdash t : \mathbb{M}$. By shrinking correctness (Thm. 3), $t \rightarrow_x^* f'_s$ for a strong fireball f'_s . By confluence of the VSC (Prop. 3.1), $f_s = f'_s$. \square

Relational Semantics. Multi types induce a denotational model, the *relational semantics*, interpreting a term as the set of its derivable judgments [18,19,25,28,6,8]. Here we focus on the semantics induced by shrinking derivations. Let t be a term and let $\vec{x} = (x_1, \dots, x_n)$ be a list of pairwise distinct variables with $n \geq 0$ and $\text{fv}(t) \subseteq \{x_1, \dots, x_n\}$: the *shrinking semantics* $\llbracket t \rrbracket_{\vec{x}}$ of t for \vec{x} is defined by $\llbracket t \rrbracket_{\vec{x}} := \{((\mathbb{N}_1, \dots, \mathbb{N}_n), \mathbb{M}) \mid \exists \text{ shrinking } \Phi \triangleright x_1 : \mathbb{N}_1, \dots, x_n : \mathbb{N}_n \vdash t : \mathbb{M}\}$.

Subject reduction and expansion (Prop. 5) guarantee that $\llbracket t \rrbracket_{\vec{x}}$ is *sound* for \rightarrow_{vsc} : if $t \rightarrow_{\text{vsc}} u$ then $\llbracket t \rrbracket_{\vec{x}} = \llbracket u \rrbracket_{\vec{x}}$. Shrinking correctness (Thm. 3) and completeness (Thm. 4), along with untyped normalization (Thm. 5), guarantee *adequacy* for this semantics, *i.e.*, they give a semantic characterization of normalization in Strong VSC: t is weakly vsc-normalizing if and only if $\llbracket t \rrbracket_{\vec{x}} \neq \emptyset$.

9 Conclusions

This paper studies call-by-value strong evaluation defined as the external strategy \rightarrow_x of the value substitution calculus (VSC). Such a strategy is analyzed using the semantical tool of Ehrhard's multi types, declined in their shrinking variant as it is standard for studying strong evaluation. The main contributions are that \rightarrow_x -normalizing terms are exactly those typable with shrinking call-by-value multi types, plus an untyped normalization theorem for \rightarrow_x in the strong VSC. These results mimic faithfully similar results for strong call-by-name.

These contributions are developed to validate the external strategy as the good notion of termination for strong call-by-value, in contrast to the other *non-equivalent* proposal in the literature given by the strong fireball calculus, which we show to terminate on some terms on which the external strategy diverges.

We conjecture that all x -normalizing terms are also normalizing in the strong fireball calculus, but a proof is likely to be very technical.

References

1. Accattoli, B.: Proof nets and the call-by-value λ -calculus. Theor. Comput. Sci. **606**, 2–24 (2015). <https://doi.org/10.1016/j.tcs.2015.08.006>

2. Accattoli, B., Condoluci, A., Sacerdoti Coen, C.: Strong Call-by-Value is Reasonable, Implausively. In: LICS 2021. IEEE (2021). <https://doi.org/10.1109/LICS52264.2021.9470630>
3. Accattoli, B., Faggian, C., Lancelot, A.: Normal form bisimulations by value. CoRR **abs/2303.08161** (2023). <https://doi.org/10.48550/arXiv.2303.08161>
4. Accattoli, B., Graham-Lengrand, S., Kesner, D.: Tight typings and split bounds. PACMPL **2**(ICFP), 94:1–94:30 (2018). <https://doi.org/10.1145/3236789>, <https://doi.org/10.1145/3236789>
5. Accattoli, B., Guerrieri, G.: Open Call-by-Value. In: APLAS 2016. Springer (2016). https://doi.org/10.1007/978-3-319-47958-3_12
6. Accattoli, B., Guerrieri, G.: Types of fireballs. In: APLAS 2018. Springer (2018). https://doi.org/10.1007/978-3-030-02768-1_3
7. Accattoli, B., Guerrieri, G.: Abstract machines for open call-by-value. Sci. Comput. Program. **184** (2019). <https://doi.org/10.1016/j.scico.2019.03.002>
8. Accattoli, B., Guerrieri, G.: The theory of call-by-value solvability. Proc. ACM Program. Lang. **6**(ICFP), 855–885 (2022). <https://doi.org/10.1145/3547652>, <https://doi.org/10.1145/3547652>
9. Accattoli, B., Guerrieri, G., Leberle, M.: Semantic bounds and strong call-by-value normalization. CoRR **abs/2104.13979** (2021), <https://arxiv.org/abs/2104.13979>
10. Accattoli, B., Guerrieri, G., Leberle, M.: Strong call-by-value and multi types (long version). CoRR **abs/2309.12261** (2023), <https://arxiv.org/abs/2309.12261>
11. Accattoli, B., Paolini, L.: Call-by-value solvability, revisited. In: FLOPS 2012. Springer (2012). https://doi.org/10.1007/978-3-642-29822-6_4
12. Accattoli, B., Sacerdoti Coen, C.: On the relative usefulness of fireballs. In: LICS 2015. IEEE (2015). <https://doi.org/10.1109/LICS.2015.23>
13. Accattoli, B., Sacerdoti Coen, C.: On the value of variables. Information and Computation **255**, 224–242 (2017). <https://doi.org/10.1016/j.ic.2017.01.003>, <https://doi.org/10.1016/j.ic.2017.01.003>
14. Biernacka, M., Biernacki, D., Charatonik, W., Drab, T.: An abstract machine for strong call by value. In: APLAS 2020 (2020). https://doi.org/10.1007/978-3-030-64437-6_8
15. Biernacka, M., Charatonik, W., Drab, T.: A derived reasonable abstract machine for strong call by value. In: PPDP 2021. ACM (2021). <https://doi.org/10.1145/3479394.3479401>
16. Bucciarelli, A., Kesner, D., Ventura, D.: Non-idempotent intersection types for the lambda-calculus. Log. J. IGPL (2017). <https://doi.org/10.1093/jigpal/jzx018>
17. Carraro, A., Guerrieri, G.: A semantical and operational account of call-by-value solvability. In: FOSSACS 2014. Springer (2014). https://doi.org/10.1007/978-3-642-54830-7_7
18. de Carvalho, D.: Sémantiques de la logique linéaire et temps de calcul. Thèse de doctorat, Université Aix-Marseille II (2007)
19. de Carvalho, D.: Execution time of λ -terms via denotational semantics and intersection types. Math. Str. in Comput. Sci. **28**(7), 1169–1203 (2018). <https://doi.org/10.1017/S0960129516000396>
20. de Carvalho, D., Pagani, M., Tortora de Falco, L.: A semantic measure of the execution time in linear logic. Theor. Comput. Sci. **412**(20), 1884–1902 (2011). <https://doi.org/10.1016/j.tcs.2010.12.017>
21. Curien, P., Herbelin, H.: The duality of computation. In: ICFP 2000 (2000). <https://doi.org/10.1145/351240.351262>
22. Dal Lago, U., Martini, S.: The weak lambda calculus as a reasonable machine. Theor. Comput. Sci. **398** (2008). <https://doi.org/10.1016/j.tcs.2008.01.044>

23. Díaz-Caro, A., Manzonetto, G., Pagani, M.: Call-by-value non-determinism in a linear logic type discipline. In: LFCS 2013. Springer (2013). https://doi.org/10.1007/978-3-642-35722-0_12
24. Dyckhoff, R., Lengrand, S.: Call-by-Value lambda-calculus and LJQ. *J. Log. Comput.* **17**(6), 1109–1134 (2007). <https://doi.org/10.1093/logcom/exm037>
25. Ehrhard, T.: Collapsing non-idempotent intersection types. In: CSL 2012. pp. 259–273. Schloss Dagstuhl (2012). <https://doi.org/10.4230/LIPIcs.CSL.2012.259>
26. Girard, J.Y.: Linear Logic. *Theoretical Computer Science* **50**, 1–102 (1987). [https://doi.org/10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4)
27. Grégoire, B., Leroy, X.: A compiled implementation of strong reduction. In: ICFP 2002. pp. 235–246. ACM (2002). <https://doi.org/10.1145/581478.581501>
28. Guerrieri, G.: Towards a semantic measure of the execution time in call-by-value λ -calculus. In: DCM/ITRS 2018 (2019). <https://doi.org/10.4204/EPTCS.293.5>
29. Guerrieri, G., Paolini, L., Ronchi Della Rocca, S.: Standardization and conservativity of a refined call-by-value lambda-calculus. *Logical Methods in Computer Science* **13**(4) (2017). [https://doi.org/10.23638/LMCS-13\(4:29\)2017](https://doi.org/10.23638/LMCS-13(4:29)2017)
30. Herbelin, H., Zimmermann, S.: An operational account of Call-by-Value Minimal and Classical λ -calculus in Natural Deduction form. In: TLCA 2009,. Springer (2009). https://doi.org/10.1007/978-3-642-02273-9_12
31. Kesner, D., Ventura, D.: Quantitative types for the linear substitution calculus. In: TCS 2014. Springer (2014). https://doi.org/10.1007/978-3-662-44602-7_23
32. Krivine, J.: Lambda-calculus, types and models. Ellis Horwood, New York (1993)
33. Levy, P.B., Power, J., Thielecke, H.: Modelling environments in call-by-value programming languages. *Inf. Comput.* **185**(2), 182–210 (2003). [https://doi.org/10.1016/S0890-5401\(03\)00088-9](https://doi.org/10.1016/S0890-5401(03)00088-9)
34. Maraist, J., Odersky, M., Turner, D.N., Wadler, P.: Call-by-name, Call-by-value, Call-by-need and the Linear λ -Calculus. *Theor. Comput. Sci.* **228**(1-2), 175–210 (1999). [https://doi.org/10.1016/S0304-3975\(98\)00358-2](https://doi.org/10.1016/S0304-3975(98)00358-2)
35. Mazza, D., Pellissier, L., Vial, P.: Polyadic approximations, fibrations and intersection types. *Proc. ACM Program. Lang.* **2**(POPL), 6:1–6:28 (2018). <https://doi.org/10.1145/3158094>
36. Moggi, E.: Computational λ -Calculus and Monads. LFCS report ECS-LFCS-88-66, University of Edinburgh (1988), <http://www.lfcs.inf.ed.ac.uk/reports/88/ECS-LFCS-88-66/ECS-LFCS-88-66.pdf>
37. Moggi, E.: Computational λ -Calculus and Monads. In: LICS '89. IEEE Computer Society (1989). <https://doi.org/10.1109/LICS.1989.39155>
38. Paolini, L.: Call-by-value separability and computability. In: ICTCS 2001 (2001). https://doi.org/10.1007/3-540-45446-2_5
39. Paolini, L., Ronchi Della Rocca, S.: Call-by-value solvability. *RAIRO Theor. Informatics Appl.* **33**(6), 507–534 (1999). <https://doi.org/10.1051/ita:1999130>
40. Plotkin, G.D.: Call-by-Name, Call-by-Value and the lambda-Calculus. *Theoretical Computer Science* (1975). [https://doi.org/10.1016/0304-3975\(75\)90017-1](https://doi.org/10.1016/0304-3975(75)90017-1)
41. Ronchi Della Rocca, S., Paolini, L.: The Parametric λ -Calculus – A Metamodel for Computation. *Texts in Theoretical Computer Science. An EATCS Series*, Springer (2004). <https://doi.org/10.1007/978-3-662-10394-4>
42. Sabry, A., Felleisen, M.: Reasoning about Programs in Continuation-Passing Style. *Lisp and Symbolic Computation* **6**(3-4), 289–360 (1993)
43. Sabry, A., Wadler, P.: A Reflection on Call-by-Value. *ACM Trans. Program. Lang. Syst.* **19**(6), 916–941 (1997). <https://doi.org/10.1145/267959.269968>
44. Terese: *Term Rewriting Systems*, Cambridge Tracts in Theoretical Computer Science, vol. 55. Cambridge University Press (2003)