Strong Call-by-Value and Multi Types

Beniamino Accattoli[1][0000−0003−4944−9944], Giulio Guerrieri[2][0000−0002−0469−4279], and Maico Leberle[1]

[1] Inria & LIX, École Polytechnique, UMR 7161, Palaiseau, France
beniamino.accattoli@inria.fr, maico-carlos.leberle@inria.fr
[2] University of Sussex, Department of Informatics, Brighton, UK
g.guerrieri@sussex.ac.uk

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.

Plotkin's call-by-value λ-calculus λ_v [32] 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 [4].

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 [31,30,33].

1. *False normal forms*: some terms are contextually equivalent to the looping term $\Omega \coloneqq (\lambda x.xx)(\lambda x.xx)$ 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 $t \to u$ then $[\![t]\!] = [\![u]\!]$) and *adequate* (that is, the denotation $[\![t]\!]$ 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 [32]. This issue was solved with categorical tools by Moggi [29], which led to a number of studies, among others [34,35,28,18,19,25], 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 [11,16,1,4,24,5,6,2,7]. 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 [21].

In [4], 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 [4].

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 [31,33], then rediscovered by Accattoli and Sacerdoti Coen [12] to study cost models, it has some similarities with a calculus introduced by Grégoire and Leroy [22] 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 [4].

Beyond False Normal Forms. In later works [5,7], 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 [20], 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 [13] and 2021 [14] 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 [14] 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 [23,5,7], 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 [27], de Carvalho [17], Kesner and Ventura [26], Bucciarelli et al. [15].

Here, we adapt to CbV the shrinking technique as presented by Accattoli et al. for CbN multi types in [3], 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 [3]. 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 [8] 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.

Paper. This work has been recently published [9]. Proofs are in [10], the long version of the paper.

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, Implosively. In: LICS 2021. IEEE (2021). https://doi.org/10.1109/LICS52264.2021.9470630
3. 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
4. Accattoli, B., Guerrieri, G.: Open Call-by-Value. In: APLAS 2016. Springer (2016). https://doi.org/10.1007/978-3-319-47958-3_12
5. Accattoli, B., Guerrieri, G.: Types of fireballs. In: APLAS 2018. Springer (2018). https://doi.org/10.1007/978-3-030-02768-1_3
6. 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
7. 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
8. 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
9. Accattoli, B., Guerrieri, G., Leberle, M.: Strong call-by-value and multi types. In: Ábrahám, E., Dubslaff, C., Tarifa, S.L.T. (eds.) Theoretical Aspects of Computing - ICTAC 2023 - 20th International Colloquium, Lima, Peru, December 4-8, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14446, pp. 196–215. Springer (2023). https://doi.org/10.1007/978-3-031-47963-2_13, https://doi.org/10.1007/978-3-031-47963-2_13
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. 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

14. 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
15. 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
16. 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
17. 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
18. Curien, P., Herbelin, H.: The duality of computation. In: ICFP 2000 (2000). https://doi.org/10.1145/351240.351262
19. 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
20. 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
21. Girard, J.Y.: Linear Logic. Theoretical Computer Science **50**, 1–102 (1987). https://doi.org/10.1016/0304-3975(87)90045-4
22. 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
23. 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
24. 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
25. 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
26. 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
27. Krivine, J.: Lambda-calculus, types and models. Ellis Horwood, New York (1993)
28. 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
29. Moggi, E.: Computational λ-Calculus and Monads. In: LICS '89. IEEE Computer Society (1989). https://doi.org/10.1109/LICS.1989.39155
30. Paolini, L.: Call-by-value separability and computability. In: ICTCS 2001 (2001). https://doi.org/10.1007/3-540-45446-2_5
31. 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
32. 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
33. 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
34. Sabry, A., Felleisen, M.: Reasoning about Programs in Continuation-Passing Style. Lisp and Symbolic Computation **6**(3-4), 289–360 (1993)
35. 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