More on modal embeddings and calling paradigms

José Espírito Santo[1], Luís Pinto[1], and Tarmo Uustalu[2,3]

[1] Centro de Matemática, Universidade do Minho, Portugal
[2] Dept. of Computer Science, Reykjavik University, Iceland
[3] Dept. of Software Science, Tallinn University of Technology, Estonia

The first connection between modal embeddings and calling paradigms in functional programming was made in the context of linear logic [7, 9]. In our work, we seek the root and the deep meaning of this connection, along the following general lines first laid out in [5]: (1) One can identify a modal calculus (a simple extension of the λ-calculus with an S4 modality) that serves as target of the modal embeddings and show that this modal target obeys a new calling paradigm, named call-by-box; (2) Two embeddings of intuitionistic logic into modal logic S4, as given in [11], where they are attributed to Girard and Gödel, can be recast as maps compiling respectively the ordinary (call-by-name, cbn) λ-calculus [1] and Plotkin's call-by-value (cbv) λ-calculus [10] into the modal target, both following the compilation technique of "protecting by a lambda", and together achieving a unification of call-by-name and call-by-value through call-by-box – all this provided Gödel's embedding suffers a slight adjustment, which, despite seeming innocent, allows a certain uniformity and simplicity in the images of the two embeddings, leading to the removal of all the β-reduction steps pertaining to the modality; (3) One can define later instantiations of the S4 modality, in the form of interpretations of the modal target into diverse other calculi, recovering by composition known embeddings like, for instance, the ones into the linear λ-calculus [9].

In the context of linear logic [9], one could already observe an asymmetry between the Girard/cbn embedding and Gödel/cbv one, with the latter enjoying slightly weaker properties. In our work [5], such an asymmetry remained: For cbn, the treatment is so neat that we may say Girard's embedding just points out an isomorphic copy of the cbn λ-calculus as a fragment of the modal target calculus. For cbv and Gödel's embedding, the results were not so satisfying. In a paper recently published [6], we investigate whether this asymmetry is inherent, or whether the modal analysis of the calling paradigms can be pushed further and reveal a hidden symmetry. We intend to present the results of this further investigation.

Recall that the modal target defined in [5] adds to the ordinary λ-calculus the term constructor $\mathsf{box}(M)$, while variables are written $\varepsilon(x)$ as a reminder of its special typing. Indeed, the type form $\Box A$ is added to simple types, contexts Γ only assign such modal types, $\mathsf{box}(M)$ has type $\Box A$ when M has type A, and $\varepsilon(x)$ has type A, if x is assigned $\Box A$. In addition, arrow types always have a modal type as antecedent. As to reduction, there is a single rule, for the contraction of β-redexes where the argument necessarily has the form $\mathsf{box}(M)$. A substitution is triggered, of M (typically of type $\Box A$) for some variable x of type A: this false mismatch is not real because there are, involved in this operation, implicit β-reduction steps of the kind engendered by the modality $\Box$.

The refinement of this modal system we propose in our new work consists in prohibiting nested modal types like $\Box\Box A$, and building into the syntax of terms T a minimum of typing information – namely the distinction between terms P, Q that can have a modal type and terms M, N that cannot. Abstractions $\lambda x.T$ and eliminations of the modality $\varepsilon(x)$ cannot have modal type, introductions of the modality $\mathsf{box}(M)$ have modal type, but applications MQ are ambiguous, so we separate two forms of them. Only now, after this separation, stemming from modal considerations alone, do we resort to an idea already employed in [5], namely the separation of a derived notion of reduction, concerning one of the forms of the application constructor. These developments create in the target system two co-existing *modes*, the "left-first" and the "right-first", with which we can qualify the application constructor and reduction. The distinction between modes turns out to be connected to the distinction between calling paradigms. We verify that the modal embeddings are characterized by the mode they give preference to: Girard's embedding translates application and reduction to left-first application and reduction, whereas Gödel's embedding prefers the right-first mode.

By refining the modal target calculus in this way and accordingly recasting the embeddings, we find out that we do not lose the neat treatment of cbn, and in addition we obtain similarly neat results for cbv, that is: Gödel's embedding becomes just the indication of an isomorphic copy of Plotkin's cbv λ-calculus as a fragment of the refined modal target. In this sense, the ordinary and Plotkin's λ-calculi, more than being unified by the original modal target, they truly co-exist inside the refined (but still simple) modal calculus.

We plan to revisit the idea of instantiation of the modality from this new standpoint. We can return to the decomposition of the embeddings into the linear λ-calculus, briefly mentioned in [5] (in this regard, the study of the bang calculus [2, 3, 4] has somewhat related goals). But we can also consider embeddings into call-by-push-value [8] in the same spirit. The thesis we would like to confirm is that calculi encompassing call-by-name and call-by-value do so because they already embed our modal target.

References

[1] H. Barendregt, The Lambda Calculus, Vol. 103 of Studies in Logic and the Foundations of Mathematics, North-Holland, 1984.

[2] A. Bucciarelli, D. Kesner, A. Ríos, A. Viso, The bang calculus revisited, in: K. Nakano, K. Sagonas (Eds.), Functional and Logic Programming: 15th Int. Symp., FLOPS 2020, Proceedings, Vol. 12073 of Lecture Notes in Computer Science, Springer, 2020, pp. 13–32.

[3] T. Ehrhard, G. Guerrieri, The bang calculus: An untyped lambda-calculus generalizing call-by-name and call-by-value, in: Proc. of 18th Int. Symp. on Principles and Practice of Declarative Programming, PPDP '16, ACM, 2016, pp. 174–187.

[4] G. Guerrieri, G. Manzonetto, The bang calculus and the two Girard's translations, in: T. Ehrhard, M. Fernández, V. de Paiva, L. T. de Falco (Eds.), Proc. of Joint Int. Workshops on Linearity and Trends in Linear Logic and Applications, Linearity-TLLA 2018, Vol. 292 of Electronic Proceedings in Theoretical Computer Science, Open Publishing Assoc., 2019, pp. 15–30.

[5] J. Espírito Santo, L. Pinto, T. Uustalu, Modal embeddings and calling paradigms, in: H. Geuvers (Ed.), 4th Int. Conf. on Formal Structures for Computation and Deduction, FSCD 2019, Vol. 131 of Leibniz Int. Proc. in Informatics, Dagstuhl Publishing, 2019, pp. 18:1–18:20.

[6] J. Espírito Santo, L. Pinto, T. Uustalu, Plotkin's call-by-value λ-calculus as a modal calculus, Journal of Logical and Algebraic Methods in Programming 127 (2022) 100775

[7] J.-Y. Girard, Linear logic, Theoretical Computer Science 50 (1) (1987) 1–102.

[8] P. B. Levy, Call-by-push-value: Decomposing call-by-value and call-by-name, Higher-Order and Symbolic Compututation 19 (4) (2006) 377–414.

[9] J. Maraist, M. Odersky, D. N. Turner, P. Wadler, Call-by-name, call-by-value, call-by-need and the linear lambda calculus, Theoretical Computer Science 228 (1–2) (1999) 175–210.

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

[11] A. Troelstra, H. Schwichtenberg, Basic Proof Theory, 2nd Edition, Vol. 43 of Cambridge Tracts in Theoretical Computer Science, Cambridge Univ. Press, 2000.