# Formalization & Computation: Categorical Normalization by Evaluation

David G. Berry[1] and Marcelo P. Fiore[2*]

Computer Laboratory, University of Cambridge

[1] `David.Berry@cl.cam.ac.uk`
[2] `Marcelo.Fiore@cl.cam.ac.uk`

This work studies a categorical presentation of normalization by evaluation formalized in the Coq proof assistant. The development followed the principle that the formalization exists to be extracted for practical computation. This philosophy motivates well the required non-standard category theory.

**Normalization and the Yoneda Embedding** Normalization by Evaluation is a technique for computing the normal forms of $\lambda$-calculi terms in a reduction-free manner. It has seen many different expositions [2, 1, 3, 4]: proof-theoretic, type-theoretic, and category-theoretic. Here, we are interested in the category-theoretic presentation of Čubrić, Dybjer, and Scott for the simply-typed $\lambda$-calculus by way of the Yoneda embedding [3]. Their starting point is noticing that there are two canonical functors from the free cartesian-closed category into its category of presheaves; *viz.*, the Yoneda embedding and the semantic interpretation functor agreeing on base-types with Yoneda. These functors are universally naturally isomorphic: a round-trip of the isomorphism is, extensionally, the identity. The insight of [3] is that by switching perspective into using an intensional variant of category theory one can derive a normalization algorithm categorically which is an instance of normalization by evaluation. This algorithm does not make use of neutral and normal terms, yet long-$\beta\eta$-normal forms are computed. Such use of the Yoneda embedding is reminiscent of coherence arguments for bicategories.

**P-Category Theory** Recently, E-category theory has received attention as a way of importing classical results into the setting provided by constructive proof assistants [6, 7]. Instead of using the in-built Martin-Löf Identity type for morphisms, E-categories come equipped with their own equivalence relation on morphisms. This turns homs into setoids, requiring operations on morphisms to respect the equivalence relation. P-category theory [3] replaces the use of equivalence relations with partial equivalence relations (*i.e.* relaxing reflexivity) and resulting in hom-subsetoids. This subtle change comes with technical advantages that provide a more appropriate setting for the categorical normalization algorithm.

The E-category of E-sets and E-maps enjoys both completeness and cocompleteness properties: equalizers can be constructed using $\Sigma$-types to restrict to the appropriate subset, and coequalizers by modifying the equivalence relation to accommodate quotienting. There is an asymmetry here: equalizers change the underlying type whereas coequalizers change the underlying equivalence relation. More importantly for our concerns, the use of $\Sigma$-types for equalizers has the undesirable side-effect of mixing formalization properties (the fibre of the $\Sigma$-type) and computational data (the base of the $\Sigma$-type). This lack of separation was noticed by Salveson and Smith [8].

The P-category of P-sets and P-maps also enjoys completeness and cocompleteness properties. However, in a symmetrical fashion: both equalizers and coequalizers are constructed by

*Research partially supported by EPSRC grant EP/V002309/1.

changing the partial equivalence relation. Crucially, this maintains good separation of formalization properties and computational data. This symmetrization results in both equalizers and coequalizers being cost-free for computation: the underlying type is unchanged.

**Coq Formalization** We have formalized the required categorical theory in the Coq proof assistant; namely, the notions of P-sets and P-categories, the P-category of P-sets and P-maps, computational ends and coends for P-sets, the notions of P-functors and P-natural transformations, the P-category of P-set-valued presheaves, the Yoneda embedding and cartesian-closed structure thereof, and the free cartesian-closed P-category with its two-dimensional universal property. This supports our synthesis of a categorical normalization-by-evaluation program.

The use of P-category theory is essential in the construction of P-categories of P-functors: the hom-subsetoids have as their underlying type all transformations, but the associated partial equivalence relation relates only those transformations which are natural. The development of our formalization has resulted in paying close attention to actually being able to compute with our formal constructions within Coq. We therefore make no use of quotients, nor assert any axioms in the Coq development.

**Future Work** The abstract categorical nature of the development should permit our aim to generalize to the bicategorical setting building on [5]. Such a categorically-developed normalization program using the bicategorical Yoneda embedding should compute normal forms for both 1-cells and 2-cells of free cartesian- closed bicategories, without the need to consider neutral and normal terms.

This work, with its distinct approach, raises questions for Univalent Foundations and Homotopy Type Theory. Specifically, can one reconcile the P-categorical approach with the UF/HoTT approach to category theory?

# References


[1] T. Altenkirch, M. Hofmann, and T. Streicher. Categorical reconstruction of a reduction free normalization proof. In D. Pitt, D. E. Rydeheard, and P. Johnstone, editors, *Category Theory and Computer Science*, pages 182–199, Berlin, Heidelberg, 1995. Springer.

[2] U. Berger and H. Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 203–211, 1991.

[3] D. Čubrić, P. Dybjer, and P. Scott. Normalization and the Yoneda embedding. *Mathematical Structures in Computer Science*, 8(2):153–192, 1998.

[4] M. Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. *Mathematical Structures in Computer Science*, 32(8):1028–1065, 2022.

[5] M. Fiore and P. Saville. Coherence for bicategorical cartesian closed structure. *Mathematical Structures in Computer Science*, 31(7):822–849, 2021.

[6] J. Gross, A. Chlipala, and D. I. Spivak. Experience implementing a performant category-theory library in Coq. In *Interactive Theorem Proving*, pages 275–291. Springer, 2014.

[7] J. Z. S. Hu and J. Carette. Formalizing category theory in Agda. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*. ACM, 2021.

[8] A. Salveson and J. Smith. The strength of the subset type in Martin-Löf's type theory. In *Proceedings Third Annual Symposium on Logic in Computer Science*, pages 384–391, 1988.