Categorical Normalization by Evaluation: A Novel Universal Property for Syntax

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 Rocq proof assistant, resulting in an effectively executable algorithm for normalizing pure λ-calculus terms. Former work on categorical normalization, such as [1, 3, 4], has failed to establish both soundness and (strong) completeness of normalization by evaluation purely categorically. Either they have relied on a non-categorical proof, or they have relied on an *a priori* characterization of normal forms to deduce some properties. In this work we present a new universal property for syntax that supports the construction of a purely categorical argument for both soundness and (strong) completeness.

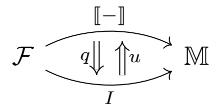
Categorical Normalization Normalization by Evaluation is a technique for computing the normal forms of λ-calculi terms in a reduction-free manner. It has seen many different expositions [2, 1, 3, 4]: proof-theoretic, type-theoretic, and category-theoretic. The category-theoretic presentation of Čubrić, Dybjer, and Scott by way of the Yoneda embedding [3] constructs a normal form algorithm purely functorially by the universal property of quotiented syntax within the setting of P-category theory. This ensures that the resultant algorithm satisfies soundness: that the output is $\beta\eta$-convertible with the input. Although, they resort to non-categorical techniques to establish that their normalization algorithm is (strongly) complete: that it maps $\beta\eta$-convertible inputs to α-equivalent outputs; and that their normalization algorithm produces β-normal–η-long–normal forms. The presentation of Fiore by way of categorical gluing [4] constructs a normal form algorithm using manually constructed maps over glued objects of neutral and normal forms. It is precisely because these forms are considered up to α-equivalence that the resultant algorithm is (strongly) complete.

Our approach combines techniques from these two prior works, with a new universal property to allow for a construction of a normal form algorithm that is both sound and (strongly) complete. This allows for an algorithm that reduces the problem of $\beta\eta$-convertibility to that of α-equivalence without needing an *a priori* known notion of normal form.

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 [5, 6]. Instead of using the in-built Martin-Löf Identity type for equality of 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), thereby resulting in hom-subsetoids. This subtle change comes with technical advantages that provide a more appropriate setting for categorical normalization algorithms. We continue in Čubrić, Dybjer, and Scott's [3] use of P-category theory to phrase our construction and formalization.

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

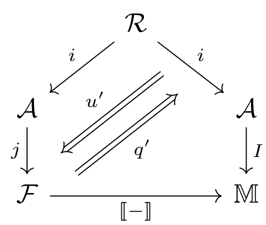
Universal Properties of Syntax The free Cartesian-closed category[1], $\mathcal{F}$, satisfies an initiality property.



For any pointed Cartesian-closed category, $\mathbb{M}$, and strictly-pointed Cartesian-closed functor, I, thereinto, there is a universal strictly-pointed Cartesian-closed interpretation functor, $[\![-]\!]$, from the free Cartesian-closed category, $\mathcal{F}$, resulting in a unique natural isomorphism, $q : [\![-]\!] \overset{\sim}{\Rightarrow} I : u$.

Čubrić, Dybjer, and Scott [3] use this fact to deduce their normal form algorithm. Moreover, their switch to P-category theory allows them to consider a second category: the category of unquotiented (*i.e.* up to α-equivalence) terms. They remark that this category is initial over some term algebra but seemingly make no use of this fact. They use presheaves over this category to deduce *non-categorically* that their algorithm is (strongly) complete. Their use of P-category theory allows them to conclude that the computational aspect of presheaves over the two categories are identical. Indeed it was a desire to replicate their argument more categorically that resulted in the presented work.

The category of unquotiented simply-typed λ-calculus terms, $\mathcal{A}$, also satisfies an initiality property but with respect to the free Cartesian-closed category, $\mathcal{F}$.



For any pointed Cartesian-closed category, $\mathbb{M}$, and strictly-pointed Cartesian-pre-closed functor, I, thereinto, there is a universal strictly-pointed Cartesian-closed interpretation functor, $[\![-]\!]$, from the free Cartesian-closed category, $\mathcal{F}$, resulting in a unique pair of natural transformations, $q' : [\![-]\!] \circ j \circ i \Rightarrow I \circ i$, and u' in the reverse direction. Where the functor j is the quotient map from unquotiented syntax to quotiented syntax, the category $\mathcal{R}$ is the category of renamings, and i is the inclusion map from renamings into (unquotiented) syntax. And Cartesian-pre-closure is a novel condition weaker than full Cartesian-closure for functors that arises from the weaker structure of $\mathcal{A}$. A category, $\mathbb{C}$, is Cartesian-pre-closed precisely when:

- it is Cartesian;
- it has a pre-exponential operator on objects:
$$(-) \Rightarrow (=) : \mathbb{C}_0 \times \mathbb{C}_0 \to \mathbb{C}_0;$$
- such that there are maps natural in c:
$$\mathbb{C}(c \times a, b) \Rightarrow \mathbb{C}(c, a \Rightarrow b)$$
$$\mathbb{C}(c, a \Rightarrow b) \Rightarrow \mathbb{C}(c \times a, b).$$

A functor, $F : \mathbb{C} \to \mathbb{D}$, with $\mathbb{C}$ Cartesian-pre-closed and $\mathbb{D}$ Cartesian-closed, is Cartesian-pre-closed precisely when:

[1] We consider freeness over a single base type in our work; one may easily instead consider freeness over a set of base types.

- it is Cartesian;
- there is a family of maps:
$$e' : F(a) \Rightarrow F(b) \to F(a \Rightarrow b);$$
- such that the following holds:
$$e' \circ (F(f) \circ p)^* \sim F(f^*).$$

By judiciously instantiating $\mathbb{M}$ and I with presheaves over $\mathcal{A}$ and the Yoneda embedding one can replicate the construction, *mutatis mutandis*, from [3] and induce a normal form algorithm that is strongly complete purely categorically. However this algorithm is not obviously sound due to the weaker naturality properties of q' and u'.

This shortfall can be overcome by further judicious instantiation of $\mathbb{M}$ with a gluing category allowing for an algorithm that is both sound and strongly complete. There are a number of possible candidates for such a gluing category that allow for the desired conclusions of soundness and (strong) completeness, each with their own resultant computational aspects. It suffices for the result to use the arrow category of presheaves over $\mathcal{A}$ for $\mathbb{M}$, and the pairing of the Yoneda embedding with the relative embedding along j for I. This realises the domain and codomain parts, respectively, as containing strict information for (strong) completeness, and weak quotient information for soundness. This gluing construction can be seen to combine aspects of [4] with our new universal property permitting the induction of a normal form algorithm purely categorically without any *a priori* knowledge of neutral and normal forms.

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] 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.

[6] 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.