A Machine-Checked Correctness Proof of Normalization by Evaluation for Simply Typed Lambda Calculus

Author: András Kovács   Advisor: Ambrus Kaposi

Budapest, 2017

Abstract

We implement and prove correct normalization by evaluation for simply typed lambda calculus, using the proof assistant Agda. The correctness proof consists of soundness, completeness and stability with respect to $\beta\eta$-conversion. The algorithm uses a Kripke model over the preordered set of order-preserving context embeddings. Completeness is given by a logical relation between the term model and the Kripke model. For soundness and stability, we first need to prove that evaluation, quoting and unquoting all commute with order-preserving embeddings, which amounts to showing that the codomain of the evaluation function can be refined to a presheaf model. Then, soundness is shown by a logical relation on semantic values which fall into the refined model. Stability is proved by induction on normal and neutral terms. Decidability of $\beta\eta$-conversion for terms also follows from correctness. The main advantage of the formalization is that normalization has a concise and structurally recursive specification and is presented separately from the correctness proofs. Also, the syntax of terms remains simple, making no use of explicit substitutions or closures. Overall, the technical overhead of the development is relatively light.

Contents

1 Introduction

This paragraph won't be part of the note, because it isn't indented. Here's a reference [1]. Here's another [3]. And yet another here [2].

More references abound [13, 9, 7, 12].

And yet more references [6, 8, 5, 4, 11, 10].

1.1 Normalization by evaluation

Lorem ipsum.

1.2 Related work

Lorem ipsum.

2 Metatheory

Lorem ipsum.

3 Syntax

Lorem ipsum.

3.1 Terms and contexts

Lorem ipsum.

3.2 Categories-with-families

Lorem ipsum.

3.3 Order-preserving embeddings

Lorem ipsum.

3.4 Substitution

Lorem ipsum.

3.4.1 Test level three header

Lorem ipsum.

3.5 Conversion

Lorem ipsum.

4 Preliminaries: normalization with a complete Kripke model

Lorem ipsum.

5 Presheaf model

Lorem ipsum.

5.1 Presheaves

Lorem ipsum.

5.2 Terms and contexts

Lorem ipsum.

5.3 Embeddings and substitutions

Lorem ipsum.

5.4 Normalization

Lorem ipsum.

6 Correctness

Lorem ipsum.

6.1 Completeness

Lorem ipsum.

6.2 Soundness

Lorem ipsum.

6.3 Stability

Lorem ipsum.

7 Discussion

Lorem ipsum.

7.1 Formal development

Lorem ipsum.

7.2 Efficiency

Lorem ipsum.

7.3 Comparison

Lorem ipsum.

7.4 Future work

Lorem ipsum.

References

[1] Thorsten Altenkirch. “A Formalization of the Strong Normalization Proof for System F in LEGO”. In: *Typed Lambda Calculi and Applications*. Ed. by J.F. Groote M. Bezem. LNCS 664. 1993, pp. 13–28.

[2] Thorsten Altenkirch. “Logical relations and inductive/coinductive types”. In: *Computer Science Logic, 12th International Workshop, CSL ’98*. LNCS 1584. 1998, pp. 343–354.

[3] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. “Categorical reconstruction of a reduction free normalization proof”. In: *Category Theory and Computer Science*. Ed. by David Pitt, David E. Rydeheard, and Peter Johnstone. LNCS 953. 1995, pp. 182–199.

[4] Thorsten Altenkirch and Ondrej Rypacek. “A Syntactical Approach to Weak omega-Groupoids”. In: *Computer Science Logic (CSL’12)*. 2012, pp. 16–30.

[5] Thorsten Altenkirch et al. “A Categorical Semantics for Inductive-Inductive Definitions”. In: *CALCO*. 2011, pp. 70–84.

[6] Thorsten Altenkirch et al. “ΠΣ: Dependent Types Without the Sugar”. In: *Functional and Logic Programming* (2010), pp. 40–55.

[7] Matt Brown and Jens Palsberg. “Self-Representation in Girard’s System U”. In: *SIGPLAN Not.* 50.1 (Jan. 2015), pp. 471–484. ISSN: 0362-1340. DOI: 10.1145/2775051.2676988.

[8] Nils Anders Danielsson and Thorsten Altenkirch. “Subtyping, declaratively; an exercise in mixed induction and coinduction”. In: *proceedings of the Tenth International Conference on Mathematics of Program Construction (MPC 10)*. 2010.

[9] Dominique Devriese and Frank Piessens. “Typed syntactic meta-programming”. In: *Proceedings of the 2013 ACM SIGPLAN International Conference on Functional Programming (ICFP 2013)*. ACM, Sept. 2013, pp. 73–85. ISBN: 978-1-4503-2326-0. DOI: 10.1145/2500365.2500575.

[10] Peter Hancock et al. “Small Induction Recursion”. In: *Typed Lambda Calculi and Applications, 11th International Conference, TLCA 2013*. 2013, pp. 156–172.

[11] Nicolai Kraus et al. “Generalizations of Hedberg’s Theorem”. In: *Typed Lambda Calculi and Applications, 11th International Conference, TLCA 2013*. 2013, pp. 173–188.

[12] Conor McBride and Ross Paterson. “Applicative programming with effects”. In: *Journal of functional programming* 18.01 (2008), pp. 1–13.

[13] The Univalent Foundations Program. *Homotopy type theory: Univalent foundations of Mathematics*. Tech. rep. Institute for Advanced Study, 2013.