Towards an implementation in LambdaProlog of the two level Minimalist Foundation

Alberto Fiori

November 3, 2017

Chapter 1

Introduction

In 2005 M. Maietti and G. Sambin [MS05b] argued about the necessity of building a foundation for constructive mathematics to be taken as a common core among relevant existing foundations in axiomatic set theory, such as Aczel-Myhill's CZF and Mizar's TG set theory, or in category theory, such as the internal theory of a topos, or in type theory, such as Martin-Lof's type theory and Coquand's Calculus of Inductive Constructions. Moreover, they asked the foundation to satisfy the "proofs-as-programs" paradigm, namely the existence of a realizability model where to extract programs from proofs. Finally, the authors wanted the theory to be appealing to standard mathematicians and therefore they wanted extensionality in the theory, e.g. to reason with quotient types and to avoid the intricacies of dependent types in intensional theories.

In the same paper they noticed that theories satisfying extensional properties, like extensionality of functions, can not satisfy the proofs-as-programs requirement. Therefore they concluded that in a proofs-as-programs theory one can only represent extensional concepts by modelling them via intensional ones in a suitable way, as already partially shown in some categorical contexts.

Finally, they ended up proposing a constructive foundation for mathematics equipped with two levels: an intensional level that acts as a programming language and is the actual proofs-as-programs theory; and an extensional level that acts as the set theory where to formalize mathematical proofs. Then, the constructivity of the whole foundation relies on the fact that the extensional level must be implemented over the intensional level, but not only this. Indeed, following Sambin's forget-restore principle, they also required that extensional concepts must be abstractions of intensional ones as result of forgetting irrelevant computational information. Such information is then restored when extensional concepts are translated back at the intensional level.

In 2009 M. Maietti [Mai09] presented the syntax and judgements of the two levels, together with a proof that a suitable completion of the intentional level provides a model of the extensional one. The proof is constructive and based on a sequence of categorical constructions, the most important being the construction of a quotient model within the intensional level, where setoids are used to encode extensional equality in an intensional language, and a notion of canonical isomorphism between intensional dependent setoids.

In this work we will present an implementation of the following software components:

1. a type checker for the intensional level

2. a reformulation of the extensional level that allows to store sintactically proof objects that are later used to provide the information to be restored when going from the extensional to the intensional level so to avoid general proof search during the interpretation

3. a type checker for the obtained extensional level

4. a translator from well-typed extensional terms to well-typed intensional terms

Combining the translator with a proof extraction component it will be possible to extract programs from proofs written in the extensional level.

We have chosen LambdaProlog [**nadatur1988LProlog**](and its recent implementation ELPI [**dunchev2015ELPI**]) as the programming language to write the two type checkers and the translator. The benefits are that LambdaProlog takes care of the intricacies of dealing with binders and alpha-conversion and moreover a LambdaProlog implementation of a syntax-directed judgement is just made of simple clauses that are almost literal translations of the judgemental rules. This allows humans (and logicians in particular) to easily inspect the code to spot possible errors.

[3] Nadathur, Gopalan, and Dale Miller. "An overview of Lambda-PROLOG." (1988).

[4] Dunchev, Cvetan, et al. "ELPI: Fast, Embeddable,\lambda Prolog Interpreter."

Bibliography

[AA15] Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Operating Systems: Three Easy Pieces.* 0.91. Arpaci-Dusseau Books, May 2015.

[Coq98] Catarina Coquand. "A realizability interpretation of Martin-Löf's type theory". In: *Twenty-Five Years of Constructive Type Theory* (1998).

[Fri75] Harvey Friedman. "Equality between functionals". In: *Logic Colloquium.* Springer. 1975, Pp. 22–37.

[GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types.* Vol. 7. Cambridge University Press Cambridge, 1989.

[Koz10] Dexter Kozen. "Church–Rosser Made Easy". In: *Fundamenta Informaticae* 103.1-4 (2010), pp. 129–136.

[Mai09] Maria Emilia Maietti. "A minimalist two-level foundation for constructive mathematics". In: *Annals of Pure and Applied Logic* 160.3 (2009), pp. 319–354.

[Mar71] P. Martin-Löf. "Hauptsatz for the intuitionistic theory of iterated inductive definitions." In: *Proceedings of the second Scandinavian logic symposium.* Ed. by J.E. Fenstad. North-Holland.

North-Holland, 1971,
Pp. 179–216.

[Mar75a] P. Martin-Löf.
"About models for intuitionistic type theories and the notion of definitional equality."
In: *Proceedings of the Third Scandinavian Logic Symposium (Univ. Uppsala, Uppsala, 1973).*
Vol. 82.
Stud. Logic Found. Math.
North-Holland, Amsterdam, 1975,
Pp. 81–109.

[Mar75b] Per Martin-Löf.
"An intuitionistic theory of types: Predicative part".
In: *Studies in Logic and the Foundations of Mathematics* 80 (1975), pp. 73–118.

[Mar98] Per Martin-Löf.
"An intuitionistic theory of types".
In: *Twenty-five years of constructive type theory* 36 (1998), pp. 127–172.

[MM16] M.E. Maietti and S. Maschio.
"A predicative variant of a realizability tripos for the Minimalist Foundation".
In: *IfCoLog Journal of Logics and their Applications* special issue Proof Truth Computation (2016).

[MS05a] M. E. Maietti and G. Sambin.
"Toward a minimalist foundation for constructive mathematics".
In: *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics.*
Ed. by L. Crosilla and P. Schuster.
Oxford Logic Guides 48.
Oxford University Press, 2005,
Pp. 91–114.

[MS05b] Maria Emilia Maietti and Giovanni Sambin.
"Toward a minimalist foundation for constructive mathematics".
In: *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics* 48 (2005), pp. 91–114.

[NPS90] Bengt Nordström, Kent Petersson, and Jan M Smith.
Programming in Martin-Löf's type theory, volume 7 of International Series of Monographs on Computer Science.

1990.