Bullet Points about Verifying Untyped Plutus Core Code

Chad E. Brown

Draft of August 12, 2021

We briefly describe a proposed research project to support formal verification of Untyped Plutus Core code [1] in an intuitionistic higher-order logic supporting higher-order abstract syntax. As preliminary work we have ported the code for the higher-order set theory proof checker Egal [4] to be a different proof checker Perche Tieni. Perche Tieni builds in Untyped Plutus Core as described in [1]. Using Perche Tieni we have verified that a term satisfies some correctness conditions and another term does not.

During the course of this preliminary work it became clear the specification in [1] in some ways differs from what is implemented in Haskell. Consequently the first thing that would need to be done to continue the project is to obtain a finalized specification of Plutus Core that corresponds to the Haskell implementation and to modify Perche Tieni to support this finalized version. After this is done, there are a number of research goals to achieve. Some of these goals are practical and would result in software to support the construction of a formal library of Untyped Plutus Core code – giving a higher degree of security than provided by type checking alone. In addition to verifying the correctness of code, one could also reason about the cost of code. In particular one could prove an alternative implementation of a function is both equivalent and consistently has a lower cost to evaluate. Some goals are theoretical and would likely result in publishable research papers. A side product of the project would likely be a series of videos explaining aspects of Untyped Plutus Core and how to reason about code written in Untyped Plutus Core.

We end with a handful of bullet points summarizing a few such goals.

- Give a formal description of the intuitionistic higher-order logic Perche

Tieni supports. This means giving the set of types, terms and propositions and characterizing the collection of theorems by giving both a proof theory (via a natural deduction [10] calculus with Curry Howard style proof terms) and a model theory (via a generalization of Henkin models [7, 2] for the intuitionistic case and supporting higher-order abstract syntax [9, 3]).

- Construction of a corresponding Proofgold Theory and support for translating conjectures and theorems (with proofs) from Perche Tieni to a format publishable in the Proofgold blockchain.[1]

- Support general automation by giving a cut-free calculus generalizing the calculus supporting the higher-order automated theorem prover Satallax to the intuitionistic case. This would involve combining the work of Brown and Smolka [5] (implemented in Satallax) with the work of Hermant and Lipton [8].

- Support specific kinds of automation, e.g., tactics for proving via induction when working with inductively defined propositions. Additionally we could apply AI-based techniques for aiding with tactic based proving, e.g., what is implemented in TacTicToe [6].

References

[1] Formal specification of the plutus core language (version 2.1), 2021.

[2] P.B. Andrews. General Models and Extensionality. *Journal of Symbolic Logic*, 37(2):395–397, 1972.

[3] Chad E. Brown. M-set models. In *Reasoning in Simple Type Theory: Festschrift in Honor of Peter B. Andrews on His 70th Birthday.* College Publications, 2008.

[4] Chad E. Brown and Karol Pąk. A tale of two set theories. In Cezary Kaliszyk, Edwin C. Brady, Andrea Kohlhase, and Claudio Sacerdoti Coen, editors, *Intelligent Computer Mathematics - 12th International*

[1] `https://proofgold.org`

Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings, volume 11617 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2019.

[5] Chad E. Brown and Gert Smolka. Analytic tableaux for simple type theory and its first-order fragment. *Logical Methods in Computer Science*, 6(2), Jun 2010.

[6] Thibault Gauthier, Cezary Kaliszyk, Josef Urban, Ramana Kumar, and Michael Norrish. Tactictoe: Learning to prove with tactics. *J. Autom. Reason.*, 65(2):257–286, 2021.

[7] Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15:81–91, 1950.

[8] Olivier Hermant and James Lipton. Cut elimination in the intuitionistictheory of types with axioms andrewriting cuts, constructively. In *Reasoning in Simple Type Theory: Festschrift in Honor of Peter B. Andrews on His 70th Birthday*. College Publications, 2008.

[9] Martin Hofmann. Semantical analysis of higher-order abstract syntax. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*, LICS '99, pages 204–213, Washington, DC, USA, 1999. IEEE Computer Society.

[10] Dag Prawitz. *Natural deduction: a proof-theoretical study.* Dover, 2006.