Matching Logic and Lean: Part 1 - Foundations

Matching logic (ML) [7, 6, 4], developed by Grigore Roşu and collaborators, is a unifying logic for defining the formal semantics of programming languages and for specifying and reasoning about the behavior of programs. Lean [9] is a functional programming language that can also be used as an interactive theorem prover [1]. The Lean project was launched by Leonardo de Moura and collaborators at Microsoft Research in 2013.

The goal of this research project is to enhance the matching logic ecosystem by adding Lean support. The following objectives are proposed.

O1 Implementation of ML in Lean

A formalization of ML in the Coq proof assistant was recently developed in [3]. Our first objective is to formalize ML in Lean. This will be similar in principle to and will be inspired from the Coq formalization, but it will be optimized and specialized to Lean, taking advantage of its strengths. We shall

- formalize the syntax, semantics, and proof system of ML;
- verify the soundness of the formalized proof system;
- formalize proofs of ML theorems and derived deduction rules, the proof of the deduction theorem, as well as other proof-theoretical tools;
- give examples of theories with proofs about their properties.

O2 Metamath proof objects

A 240-line proof checker for ML was formalized in [3, 8] using Metamath [5, 10], a computer language for mathematical proofs that is simple, fast, and trustworthy. We shall combine the Lean and Metamath formalizations of ML and translate ML proofs in Lean to proof objects in Metamath. In this way

- we shall increase the confidence in the correctness of the ML proofs;
- proof checking will be much more efficient, as it will be done by a very small proof checker.

Ultimately, this will offer the $\mathbb{K}$ ecosystem a trusted interactive formal verification environment, where Lean, $\mathbb{K}$, or any other complex system, are eliminated from the trust base.

References

[1] J. Avigad, L. de Moura, S. Kong, S. Ullrich. Theorem Proving in Lean 4. https://leanprover.github.io/theorem_proving_in_lean4/.

[2] P. Bereczky, X. Chen, D. Horpácsi, L. Peña, and J. Tušil. Mechanizing Matching Logic In Coq. arXiv:2201.05716 [cs.LO] (2022).

[3] X. Chen, Z. Lin, M. Trinh, G. Roşu. Towards a trustworthy semantics-based language framework via proof generation. In: A. Silva, K.R.M. Leino (eds.) Computer Aided Verification - 33rd International Conference, CAV 2021, Virtual Event, July 20-23, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12760, pp. 477–499. Springer (2021).

[4] X. Chen, G. Roşu. Matching μ-logic. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019. pp. 1–13. IEEE (2019).

[5] N. Megill, D.A. Wheeler. Metamath: a computer language for mathematical proofs. Lulu.com (2019).

[6] G. Roşu. Matching logic. Logical Methods in Computer Science 13:1-61 (2017).

[7] G. Roşu, C. Ellison, W. Schulte. Matching Logic: An Alternative to Hoare/Floyd Logic. AMAST:142-162 (2010).

[8] K Team. Matching logic proof checker. GitHub page. https://github.com/kframework/matching-logic-proof-checker.

[9] Lean Theorem Prover. https://leanprover.github.io.

[10] Metamath Home Page. http://us.metamath.org.