

MASTER THESIS PROPOSAL
COMPUTING SCIENCE



RADBOD UNIVERSITY

**Extending Iris with Inductive
predicates using Elpi**

Author:

Luko van der Maas
Computer science
luko.vandermaas@ru.nl
s1010320

Supervisor:

dr. Robbert Krebbers
robbert@cs.ru.nl

Assessor:

dr. Freek Wiedijk
freek@cs.ru.nl

March 27, 2024

Start date: 01-September-2023

Expected end date: 01-June-2024

Problem statement

In the Iris separation logic [Jun+15; Jun+16; Kre+17; Jun+18], implemented in Coq as MoSeL [KTB17; Kre+18], there has long been a need to represent recursive data structures in memory using mathematical structures. Several techniques have been created to solve this problem. When the recursive structure of the data structure mimics the inductive structure of the mathematical structure used to represent it, we can use the Coq fixpoint mechanism to define our representation predicate. However, when this is not the case there are three different manual ways of creating a representation predicate where available. This leads to a higher level of understanding needed to do simple tasks and more laborious steps in proofs.

Recently, a new meta language for Coq has been developed in order to create new tactics and commands. This meta language is called Elpi with its connector, Coq-Elpi [Dun+15; GCT19]. Using Elpi we can automatically create definitions and proofs and create tactics for the interactive prover.

The problem statement of this proposal is:

Is it possible, from an inductive definition, to automate the definition of a representation predicate using Elpi?

Included in the definition are the predicate itself and any associated lemmas such as unfold and induction lemmas. Also included are tactics for easily applying these lemmas.

Approach

I will be trying to solve the problem as stated above by creating several commands and tactics in order to automate the creation of a representation predicate and simplify the use of them in a proof. This will consist of several Elpi programs together with the needed Coq code in order to create these commands and tactics.

Roadmap

I will start by deepening my understanding of Iris and MoSeL together while reading more about Elpi. As a proof of concept I hope to implement a simpler tactic already existent in MoSeL and use this knowledge in order to create a plan for the commands and tactics needed for the representation predicate. This will probably take the first few months. Next I will try to

implement the commands according to the roadmap created. During this time I will start writing my thesis about the previous section of work. Lastly, I will write the rest of the thesis.

Literature

Relevant literature includes the literature about Iris and its proof mode MoSeL [Jun+15; Jun+16; Kre+17; Jun+18; KTB17; Kre+18; Iri23]. These describe how the Iris logic works and how it is implemented in Coq. Secondly we have the literature about the Elpi programming language and the Elpi-Coq connector [Dun+15; GCT19]. These explain what Elpi is and how it is used together with Coq. They also describe how a type checker was written using Elpi. Lastly, to create a representation predicate we will most likely make use least fixpoints. These are originally described by Knaster and Tarski [Tar55] and very useful for creating inductive predicates.

Bibliography

- [Dun+15] Cvetan Dunchev et al. “ELPI: Fast, Embeddable, λ Prolog Interpreter”. In: *Log. Program. Artif. Intell. Reason.* Lecture Notes in Computer Science. 2015, pp. 460–468. DOI: 10.1007/978-3-662-48899-7_32.
- [GCT19] Ferruccio Guidi, Claudio Sacerdoti Coen, and Enrico Tassi. “Implementing Type Theory in Higher Order Constraint Logic Programming”. In: *Math. Struct. Comput. Sci.* 29.8 (Sept. 2019), pp. 1125–1150. DOI: 10.1017/S0960129518000427.
- [Iri23] The Iris Team. “The Iris 4.1 Reference”. In: (Oct. 11, 2023), pp. 51–56. URL: <https://plv.mpi-sws.org/iris/appendix-4.1.pdf>.
- [Jun+15] Ralf Jung et al. “Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning”. In: *Proc. 42nd Annu. ACM SIGPLAN-SIGACT Symp. Princ. Program. Lang.* POPL ’15. Jan. 14, 2015, pp. 637–650. DOI: 10.1145/2676726.2676980.
- [Jun+16] Ralf Jung et al. “Higher-Order Ghost State”. In: *SIGPLAN Not.* 51.9 (Sept. 4, 2016), pp. 256–269. DOI: 10.1145/3022670.2951943.
- [Jun+18] Ralf Jung et al. “Iris from the Ground up: A Modular Foundation for Higher-Order Concurrent Separation Logic”. In: *J. Funct. Program.* 28 (Jan. 2018), e20. DOI: 10.1017/S0956796818000151.
- [Kre+17] Robbert Krebbers et al. “The Essence of Higher-Order Concurrent Separation Logic”. In: *Program. Lang. Syst.* Lecture Notes in Computer Science. 2017, pp. 696–723. DOI: 10.1007/978-3-662-54434-1_26.
- [Kre+18] Robbert Krebbers et al. “MoSeL: A General, Extensible Modal Framework for Interactive Proofs in Separation Logic”. In: *Proc. ACM Program. Lang.* 2 (ICFP July 30, 2018), 77:1–77:30. DOI: 10.1145/3236772.

- [KTB17] Robbert Krebbers, Amin Timany, and Lars Birkedal. “Interactive Proofs in Higher-Order Concurrent Separation Logic”. In: *SIGPLAN Not.* 52.1 (Jan. 1, 2017), pp. 205–217. DOI: 10.1145/3093333.3009855.
- [Tar55] Alfred Tarski. “A Lattice-Theoretical Fixpoint Theorem and Its Applications”. In: *Pac. J. Math.* 5.2 (June 1, 1955), pp. 285–309. URL: <https://msp.org/pjm/1955/5-2/p11.xhtml>.