Engineering Safe AI Models with Dependently Typed Languages

Contacts

Humbert Fiorino (Humbert.Fiorino@imag.fr) LIG-Marvin, Damien Pellier (Damien.Pellier@imag.fr) LIG-Marvin

Keywords

Automated planning, PDDL language, dependently typed languages, safety, provable AI.

Context

Automated planning [1] is a field of the Artificial Intelligence whose purpose is to devise decision algorithms for autonomous systems i.e., agents such as robots, drones, bots etc. As these systems act without human supervision, that is "autonomously", they have to generate plans of actions at all times in order to achieve the goals that have been assigned to them. Autonomous planning is known to be NP-hard and Domain-Specific Languages (DSL) like PDDL (Planning Domain Description Language) [3] and HDDL [4] have been designed to model/specify agent missions (the actions, goals and world states etc.) as planning problems. How plans are generated by agents is out of the scope of this research project (see PDDL4J [5] and [1] for more details). The point is that PDDL/HDDL models are hand-coded by error-prone human experts. And as, for instance, Human-Robot Interactions involve human operators sharing spaces with machines making autonomous decisions, safety and correctness in the specifications of the agent missions have become a major concern for PDDL/HDDL developers.

On the other hand, with dependently typed languages such as Coq, Agda, Lean or Idris [2], developers have the option to add more expressiveness to their code by extending the expressiveness of types and consequently the compilers can catch more errors and bugs. Thanks to the Curry-Howard equivalence, these languages are often used for formal verification and mathematical reasoning on world state reachability analysis etc.

Objective

It is well known that debugging and maintaining high level symbolic code like PDDL/HDDL models is a difficult task. Unlike other modelling languages, PDDL/HDDL lacks suitable tools for model validation/verification and proofs. The purpose of this research project is to investigate how to encode PDDL/HDDL models into provable/verifiable models. Due to their logical proximity with PDDL/HDDL, the targeted languages are programming languages with dependent types.

Expected results

The expect results are the following:

- A state-of-the-art bibliography on model validation based on type checking,
- A method to encode (a fraction of) PDDL/HDDL into (a fraction of) a dependently typed language such as Idris [2],
- A parser from PDDL to this dependently typed language,
- An evaluation of this parser based on PDDL/HDDL benchmarks.

References

[1] M. Ghallab, D. Nau and P. Traverso, "Automated Planning", Morgan-Kaufman, 2004.

[2] E. Brady, "Type-Driven Development with Idris", Manning, 2017.

[3] M. Fox, D. Long, "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains", J. Artif. Intell. Res. 20: 61-124, 2003.

[4] D. Höller, G. Behnke, P. Bercher, S. Biundo, H. Fiorino, D. Pellier, R. Alford. *HDDL,* "A Language to Describe Hierarchical Planning Problems". In the proceedings of the Conference on Artificial Intelligence (AAAI), 2020.

[5] D. Pellier, H. Fiorino, "PDDL4J: A Planning Domain Description Library for Java", Journal of Experimental & Theoretical Artificial Intelligence, pages 143-176, volume 30(1), 2018.