

My Experience with the Lean Theorem Prover

Jakob von Raumer

University of Nottingham, Nottingham, United Kingdom
psxjv4@nottingham.ac.uk

Lean is a dependently typed theorem prover¹ which is developed mainly by Leonardo de Moura at Microsoft Research. Even though the project is very young – the development started in 2013 – it already provides a very usable language for formalizations of all kinds. It allows for proof irrelevant reasoning as well as homotopy type theory and it has an extensive library for both of these modes which is mainly written by Jeremy Avigad and others at Carnegie Mellon University, Pittsburgh.

I will first give a little tour through the features of Lean, presenting its syntax and current means of automation, and comparing it to other widespread provers like Coq, Agda and Isabelle. The features presented include:

- Defining inductive types,
- defining and extending structures (record types),
- writing proof terms in direct syntax or using tactics, and
- using type classes.

Then, I will show in some examples how these features were used to build Lean's (proof irrelevant) standard library and its homotopy type theory library.

Afterwards, I will describe my experience formalizing double groupoids with thin structure and crossed modules as algebraic structures to contain the first and second homotopy groupoids of a higher type. I will focus on the difficulties to reconcile abstraction height and proof checker performance, which can be an issue in any larger formalization project.

¹<http://leanprover.github.io>