Interactive proofs with Dedukti

- Lab: LSV, ENS Cachan, France
- Team: Deducteam
- Advisor: Frédéric Blanqui (INRIA)

Dedukti is a formal proof checker based on a logical framework called the λΠ-calculus modulo, which is an extension of the simply-typed lambda-calculus with dependent types (e.g. vectors, matrices) and an equivalence relation on types generated by the user-defined rewrite rules. Proofs generated by some automated theorem provers (e.g. Zenon, iProver) or proof assistants (e.g. HOL, Coq, Matita) can be checked in Dedukti by encoding function definitions and axioms by rewrite rules [4].

But, currently, no proof assistant fully uses all the features of Dedukti, which allows arbitrary user-defined rewrite rules (e.g. \((x + y) + z \rightarrow x + (y + z)\), where + is itself defined by other rules). Such rules are indeed necessary if one wants to ease the use of dependent types and, for instance, be able to define types for representing simplicial sets of arbitrary dimensions, ∞-categories or models of Voevodsky’s homotopy type theory.

The goal of this internship is to develop an interface for Dedukti, that is, a tool for enabling users to build Dedukti proofs interactively. Two approaches are possible: developing a plugin for a well-known text editor (e.g. Emacs [2], jEdit [8], Eclipse [5]) or a JavaScript application that can be run into any browser [6, 7, 1]. In both cases, a protocol needs to be defined and implemented for the interface (editor or browser) to communicate with the proof checker. The student will have to consider to adapt one of the two main protocols: the one of ProofGeneral [3] that has been adapted to various provers already (Isabelle, Coq, PhoX, HOL, etc.), or PIDE initially developed for Isabelle [10] and recently adapted to Coq too [9] [5] [1].

Expected abilities: knowledge of LISP or JavaScript.

References


