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 $\lambda\Pi$-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. $(x+y)+z \to x+(y+z)$, 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

[1] E. J. Gallego Arias, B. Pin, and P. Jouvelot. jsCoq: towards hybrid theorem proving interfaces. In *Proceedings of the 12th Workshop on User Interfaces for Theorem Provers*, Electronic Proceedings in Theoretical Computer Science ?, 2016.

[2] D. Aspinall. Proof General: a generic tool for proof development. In *Proceedings of the 6th International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 1785, 2000.

[3] D. Aspinall, C. LÃijth, and D. Winterstein. A framework for interactive proof. In *Proceedings of the 6th International Conference on Mathematical Knowledge Management*, Lecture Notes in Computer Science 4573, 2007.

[4] A. Assaf, G. Burel, R. Cauderlier, D. Delahaye, G. Dowek, C. Dubois, F. Gilbert, P. Halmagrand, O. Hermant, and R. Saillard. Expressing Theories in the lambda-Pi-Calculus Modulo Theory and in the Dedukti System, 2016. Draft.

[5] A. Faithfull, J. Bengtson, E. Tassi, and C. Tankink. Coqoon - an IDE for interactive proof development in Coq. In *Proceedings of the 22nd International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 9636, 2016.

[6] C. Kaliszyk. Web interfaces for proof assistants. In *Proceedings of the 7th Workshop on User Interfaces for Theorem Provers*, Electronic Notes in Theoretical Computer Science 174(2), 2006.

[7] C. Lüth and M. Ring. A web interface for Isabelle: the next generation. In *Proceedings of the 6th International Conference on Intelligent Computer Mathematics*, Lecture Notes in Computer Science 7961, 2013.

[8] M. Wenzel. Isabelle/jEdit – a prover IDE within the PIDE framework. In *Proceedings of the 5th International Conference on Intelligent Computer Mathematics*, Lecture Notes in Computer Science 7362, 2012.

[9] M. Wenzel. PIDE as front-end technology for Coq, 2013.

[10] M. Wenzel. Asynchronous User Interaction and Tool Integration in Isabelle/PIDE. In *Proceedings of the 5th International Conference on Interactive Theorem Proving*, Lecture Notes in Computer Science 8558, 2014.