Efficient and modular higher-order rewriting

F. Blanqui (INRIA) and O. Hermant (MINES ParisTech)

Context. Dedukti is a universal proof checker developed at INRIA in Deducteam [2]. It has been designed so as to be small (2000 lines of OCaml) but powerful enough to express and check formal proofs coming from various automated theorem provers (e.g. Zenon, iProver) and interactive proof assistants (e.g. Coq, Matita, HOL). It is like a logical framework with the additional advantage of preserving computations by allowing shallow embeddings. The current Dedukti library contains hundreds of Mo of proofs, hence efficiency is important.

Bindlib is an OCaml library developed at the University of Chambéry providing efficient data structures for manipulating terms with bound variables, used in the programming language PML. Some preliminary experience shows that a re-implementation of Dedukti based on Bindlib could be faster, simpler and easier to extend.

Subject. The main objective of this internship will be to develop and test a new implementation of Dedukti based on Bindlib. Two approaches are possible: directly modify Dedukti's code to replace some of its data structures by the ones of Bindlib, or else start from a small prototype and extend it step-by-step to a full implementation of $\lambda\Pi$-calcul modulo, the underlying Dedukti framework [3, 1]. This achievement may lead to a publication.

Depending on his/her skills and preferences, the candidate could then extend Dedukti to more general forms of computations (e.g. rewriting with matching modulo associativity and commutativity), or the development of an interface for building Dedukti proofs interactively.

Some basic knowledge of functional programming or lambda-calculus is welcome.

Organization. The internship will take place at LSV, but the candidate could start by one or two weeks with Christophe Raffalli at Chambéry to learn how to use Bindlib.

References

[1] A. Assaf. *A framework for defining computational higher-order logics*. PhD thesis, École Polytechnique, France, 2015.

[2] Deducteam. Expressing Theories in the lambda-Pi-Calculus Modulo Theory and in the Dedukti System, 2016. Draft.

[3] R. Saillard. *Type Checking in the Lambda-Pi-Calculus Modulo: Theory and Practice*. PhD thesis, Mines ParisTech, France, 2015.