

# Structured proofs in Dedukti

Frédéric Blanqui (INRIA)  
frederic.blanqui@inria.fr

**Place:** Deducteam, Laboratoire Spécification et Vérification (LSV), ENS Paris-Saclay, 4 avenue des Sciences, Gif-sur-Yvette, France.

**Context:** Dedukti v3 aka Lambdapi (<https://github.com/Deducteam/lambdapi>) is a proof assistant, that is, a tool allowing to build and check mathematical proofs formally. Dedukti is also a logical framework, that is, it does not come with a pre-defined logic: the user has to define it, which is easy [1]. Yet, Dedukti v3 provides basic proof tactics that can be used in various user-defined logics. However, there is currently nothing to structure those proofs which makes their reading and maintenance difficult and fragile.

**Objectives:** The objective of this internship is to design and implement in OCaml a language for structuring proof scripts in Dedukti v3. Proofs inherently have a tree structure. For instance, to prove that a property  $P(n)$  holds for all natural numbers  $n$ , one can proceed by induction on  $n$  (tactic **induction**). To this end, one first needs to prove  $P(0)$  and also that  $\forall n, P(n) \Rightarrow P(n+1)$ , which makes two subgoals, and the proof of each subgoal may need the application of further proof tactics. It is therefore important to know, among the following tactics given by the user, what are those corresponding to the first subgoal and those corresponding to the second subgoal.

**Workplan:** If the intern is not familiar with the use of proof assistants, he will start by getting some familiarity with current proof assistants (Coq and Lambdapi) by reading some textbooks and manuals, and making simple proofs [3]. The intern will then read a few articles describing various approaches to proof languages [4, 5, 2], and compare them. Based on this study, the intern will propose a proof language for Dedukti v3. After validation, the intern will implement it, and do some experiment with it.

**Requirements:** Basic knowledge of logic and functional programming.

## References

- [1] A. Assaf, G. Burel, R. Cauderlier, D. Delahaye, G. Dowek, C. Dubois, F. Gilbert, P. Halmagrand, O. Hermant, and R. Saillard. Dedukti: a logical framework based on the  $\lambda\Pi$ -calculus modulo theory, 2019. Draft.

- [2] L. Lamport. How to Write a 21st Century Proof. *Journal of Fixed Point Theory and Applications*, 11:43–63, 2012.
- [3] B. C. Pierce, A. Azevedo de Amorim, C. Casinghino, M. Gaboardi, M. Greenberg, tcu C. Hri V. Sjöberg, and B. Yorgey. *Logical Foundations*, volume 1 of *Software Foundations*. Electronic textbook, 2020. version 5.8.
- [4] M. Wenzel. Isar - a generic interpretative approach to readable formal proof documents. In *Proceedings of the 12th International Conference on Theorem Proving in Higher Order Logics*, Lecture Notes in Computer Science 1690, 1999.
- [5] M. Wenzel and F. Wiedijk. A comparison of the mathematical proof languages Mizar and Isar. *Journal of Automated Reasoning*, 29:389–411, 2002.