# Modern user interface for interactive theorem proving

- Lab: LSV, Cachan, France
- Team: Deducteam
- Advisor: Frédéric Blanqui (INRIA) and Emilio Gallego (Mines ParisTech)

**Context.** Lambdapi is a new proof assistant based on a logical framework called the $\lambda\Pi$-calculus modulo rewriting, which is an extension of the simply-typed $\lambda$-calculus (the basis of functional programming languages like OCaml or Haskell) with dependent types (e.g. vectors and matrices of some given dimension) and an equivalence relation on types generated by user-defined rewrite rules [1]. Thanks to rewriting, Lambdapi allows the formalization of proofs that cannot be done in other proof assistants (e.g. simplicial sets of infinite dimensions).

However, for developing large proofs, it is essential to have a good interface. Interactive theorem proving is built on the mutual interaction between a human and a prover. The human will submit a candidate proof, and the proof assistant will confirm or reject the user proposal. Building large proofs is a very difficult task, and users do require large amount of help from the tools. Searching, completion, project management, are all essential to the successful development of large proofs.

**Goal.** The goal of the internship is to develop a modern, standard-based interface for interactive theorem proving. The interface should provide users with good capabilities for the development of proof documents, including standard editing facilities such as completion, outlines and context-aware help, with a particular focus on the proof development process which is mainly based in a challenge-response system from the prover to the user.

In particular, the student will work in the "language server paradigm". In this setting the theorem prover provides a special server that responds to queries from the editors. Then, the role of the student is to develop a plugin for a state-of-the-art editor that communicates with the language server and helps the user to develop proof documents.

In the context of this internship the student will work with the Lambdapi theorem prover. Lambdapi already provides a language server based on the Language Server Protocol (LSP) standard [2, 3, 4].

For the editor we offer two choices: VSCode, a popular visual editor or Emacs, which is textual based and popularized in the theorem proving community by Proof General.

**Workplan.**

- syntax coloring
- enable Unicode symbols
- display error messages and their locations
- display unsolved goals
- for an unsolved goal, display the assumptions
- provide buttons and short cuts for going forward or backward in a proof
- display informations about symbols or text selections (e.g. type, definition, rewrite rules)
- propose to LSP developers an extension of LSP for interactive proof development

**Requirements.** Knowledge of JavaScript and JSON. Knowledge of LISP if one wants to develop the Emacs plugin.

# 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, 2016. Draft.

[2] E. J. Gallego-Arias. SerAPI: Machine-Friendly, Data-Centric Serialization for Coq, 2016.

[3] 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*, 2016.

[4] I. Lachheb. Une interface pour Dedukti. `https://hal.inria.fr/hal-01898401`, 2018. Internship report.