

Importing Logipedia proofs in Agda

Frédéric Blanqui
frederic.blanqui@inria.fr

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

Context: This internship is about formal proofs, more specifically their translation between various proof systems.

Formal proofs are used in mathematics but also in the industry for certifying the correctness of protocols, software and hardware.

Interoperability is a very important feature in computer science and engineering to avoid useless work duplication and more safety. Unfortunately, interoperability between proof systems is not well developed because it can only be partial. Indeed, proof systems sometimes have incompatible features: combining them make them inconsistent. Therefore, to translate a proof from one system to another one, we first need to analyze the features of the first system used in the proof and check whether they are compatible with the features of the target system.

Logipedia is a prototype of online encyclopedia of mathematical theorems whose proofs are provided in many different proof languages (Coq, OpenTheory, Lean, Matita, PVS) because they have first been expressed in a simpler logical system called STTfa for Simply Type Theory with Prenex Polymorphism, an intuitionist version of higher-order logic as implemented in HOL-Light, HOL4 or Isabelle/HOL.

Objective: This internship aims at extending Logipedia to Agda, that is, at translating STTfa proofs into Agda proofs.

The main difficulty is that Agda is a predicative proof system, that is, each Agda object has to be defined in some fixed universe $k \in N$, the type of which is in the universe $k + 1$, and objects in universes k_1, \dots, k_n can be combined to craft new objects but those new objects can only live in the universe $\max(k_1, \dots, k_n)$.

Hence, to translate STTfa proofs to Agda, one must first determine in which Agda universe each STTfa object has to be translated.

Workplan: The intern will start by getting some familiarity with STTfa [1, 3, 4] and Agda. To solve the problem of universes, the intern could use a tool called Universo that allows to solve constraints on universes using SMT solvers.

Requirements: Basic knowledge of logic and functional programming (Logi-
pedia is written in OCaml).

Possible followings: If time permits, the intern could extend the previous
works on the reverse translation, that is, from Agda to Dedukti and STTfa [2].

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] G. Genestier. Encoding Agda Programs Using Rewriting. In *Proceedings of the 5th International Conference on Formal Structures for Computation and Deduction*, Leibniz International Proceedings in Informatics 167, 2020.
- [3] F. Thiré. Sharing a Library between Proof Assistants: Reaching out to the HOL Family. In *Proceedings of the 13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice*, Electronic Proceedings in Theoretical Computer Science 274, 2018.
- [4] F. Thiré. Cumulative types systems and levels. <https://hal.archives-ouvertes.fr/hal-02150179>, 2019. Presented at LFMTTP'19.