



HAL
open science

Logipedia: a multi-system encyclopedia of formal proofs

Gilles Dowek, François Thiré

► **To cite this version:**

Gilles Dowek, François Thiré. Logipedia: a multi-system encyclopedia of formal proofs. 2020. hal-04060141

HAL Id: hal-04060141

<https://inria.hal.science/hal-04060141>

Preprint submitted on 6 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LOGIPEDIA: a multi-system encyclopedia of formal proofs

Gilles Dowek and François Thiré

Inria and École normale supérieure de Paris-Saclay
LSV, 61, avenue du Président Wilson 94235 Cachan Cedex, France
`gilles.dowek@ens-paris-saclay.fr`
`francois.thire@lsv.fr`

Libraries of formal proofs are an important part of our mathematical heritage, but their usability and sustainability is poor. Indeed, each library is specific to a proof system, sometimes even to some version of this system. Thus, a library developed in one system cannot, in general, be used in another and when the system is no more maintained, the library may be lost. This impossibility of using a proof developed in one system in another has been noted for long and a remediation has been proposed: as we have empirical evidence that most of the formal proofs developed in one of these systems can also be developed in another, we can develop a standard language, in which these proofs can be translated, and then used in all systems supporting this standard.

LOGIPEDIA (<http://logipedia.science>) is an attempt to build such a multi-system on-line encyclopedia of formal proofs expressed in such as standard language. It is based on two main ideas: the use of a logical framework and of reverse mathematics.

Logical frameworks Different proof systems, such as COQ, MATITA, HOL LIGHT, ISABELLE/HOL, PVS... implement variants of different *logical formalisms*: the Calculus of constructions, Simple type theory, Simple type theory with predicate subtyping... After several decades of research, we understand the relationship between these formalisms much better. But, to build an encyclopedia of formal proofs, we have been one step further and expressed all these formalisms as theories in a common *logical framework*.

The idea of using a logical framework, such as predicate logic, to express theories, such as geometry and set theory, goes back to Hilbert and Ackermann, but several other logical frameworks such as *λ -Prolog*, *Isabelle*, *the $\lambda\Pi$ -calculus deduction modulo theory*, and *the $\lambda\Pi$ -calculus modulo theory* [3, 5], have been proposed to solve some issues of predicate logic. We have used the *the $\lambda\Pi$ -calculus modulo theory*, implemented in the system DEDUKTI [2]. With respect to predicate logic this logical framework allows

- binders (like *λ -Prolog*, *Isabelle*, and *the $\lambda\Pi$ -calculus*)
- proofs as λ -terms (like *the $\lambda\Pi$ -calculus*)
- an arbitrary definitional / computational equality (like *deduction modulo theory*)
- arbitrary connectives and quantifiers, in particular a mix of constructive classical ones (like ecumenical logics [6]).

It permits to express Simple type theory with 8 symbol declarations and 3 rewrite rules and the Calculus of constructions with 9 symbol declarations and 4 rewrite rules [4].

Expressing the theory implemented in HOL LIGHT and MATITA theories D[HOLL] and D[Mat] in DEDUKTI allows to translate proofs developed in these systems to these theories and back.

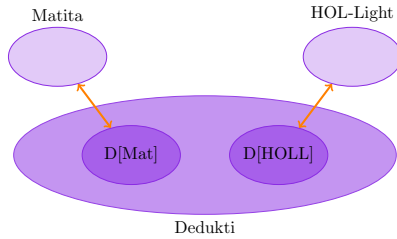


Figure 1: Logical Framework

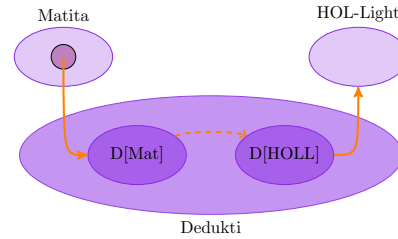


Figure 2: Reverse mathematics

Reverse mathematics Expressing various formalisms as theories in the same logical framework permits to compare them. For instance comparing the expression of Simple type theory and of the Calculus of constructions in DEDUKTI [4], we notice that there are only three differences the symbol *arrow* that permits to build functional types is dependent in the Calculus of constructions, and the same holds for the implication \Rightarrow . Then the Calculus of constructions has one extra symbol π to build types for functions mapping proofs to terms and Simple type theory does not. Thus all proofs expressed in D[HOLL] can be translated to D[Mat].

Conversely, we can identify a subset of the proofs of D[Mat], that do not use the dependency of the symbols *arrow* and \Rightarrow and do not use the symbol π , and can be translated to D[HOLL]. Composing these translation we obtain a partial function mapping MATITA proofs to HOL LIGHT PROOFS.

Instead of keeping proofs in various libraries such that that of HOL LIGHT or MATITA, and use translators, we can as well build a single multi-system encyclopedia such as LOGIPEDIA and use them directly when developing a proof in one system or the other.

Empirical results As a proof of concept, we have translated to D[Mat] the arithmetic library of MATITA [1] up to Fermat’s little theorem (around 300 lemmas). We have then translated it to D[HOLL] and exported it to HOL LIGHT, ISABELLE/HOL, COQ, LEAN, PVS, and, of course, MATITA.

The proofs are available at <http://logipedia.science>.

References

- [1] Ali Assaf. *A framework for defining computational higher-order logics. (Un cadre de définition de logiques calculatoires d’ordre supérieur)*. PhD thesis, École Polytechnique, Palaiseau, France, 2015.
- [2] Ali Assaf, Guillaume Burel, Raphaël Cauderlier, David Delahaye, Gilles Dowek, Catherine Dubois, Frédéric Gilbert, Pierre Halmagrand, Olivier Hermant, and Ronan Saillard. Dedukti: a logical framework based on the lambda-Pi-calculus modulo theory. Submitted to publication.
- [3] Denis Cousineau and Gilles Dowek. Embedding pure type systems in the lambda-pi-calculus modulo. In *Typed Lambda Calculi and Applications, 8th International Conference, TLCA 2007, Proceedings, 2007*.
- [4] Gilles Dowek. Analyzing individual proofs as the basis of interoperability between proof systems. In *Proceedings of the Fifth Workshop on Proof eXchange for Theorem Proving, PxTP 2017.*, 2017.
- [5] Bengt Nordström, Kent Petersson, and Jan M. Smith. Martin-Löf’s type theory. In *Handbook of Logic in Computer Science*. 2000.
- [6] Luiz Carlos Pereira and Ricardo Rodriguez. Normalization, soundness and completeness for the propositional fragment of Prawitz’ ecumenical system. *Revista Portuguesa de Filosofia*, 2017.