Higher Order Proof Engineering: Proof Collaboration, Transformation, Checking and Retrieval

Shuai Wang

Higher Order Proof Engineering: Proof Collaboration, Transformation, Checking and Retrieval

Shuai Wang*

ILLC, University of Amsterdam
The Netherlands
shuai.wang@student.uva.nl

Abstract

Higher Order Logic has been used in formal mathematics, software verification and hardware verification over the past decades. Recent developments of interactive theorem made sharing proofs between some theorem provers possible. This paper first gives an introduction and an overview of related recent advances, followed by the proof checking benchmarks of a proof sharing repository, namely OpenTheory (after proof transformation by the upgraded Holide). Finally, we introduce ProofCloud, the first proof retrieval engine for higher order proofs.

1 Introduction to Higher Order Proof Engineering

Higher order logic is also known as simple type theory. It is an extension of simply typed λ-calculus with additional axioms and inference rules [4]. Interactive Theorem Provers (ITPs) of higher order logic have been playing an important role in software verification, hardware verification and formal mathematics. Among them, the HOL family consists of HOL Light [8], HOL4 [13] and ProofPower [11], etc. These ITPs implemented the same higher order logic, namely Church's simple type theory [9]. However, they each contain significant theory formalizations that are not accessible to each other. HOL Light [8] has a formalization of complex analyses while HOL4 has a formalization of probability theory [13]. In contrast, ProofPower has a formalization of the Z specification language [11]. Proof libraries from one ITP may contribute to the proof automation of another ITP [7, 5]. For the sake of proof sharing between ITPs with different theories, OpenTheory [9] was developed as a cross-platform proof package manager for proofs in the HOL family. It consists of a standard library and many proof packages including lists, natural numbers, functions, etc. Taking advantage of these packages, OpenTheory has inspired further exploration of theory management [10, 6] as well as the development of several projects [1, 2, 14].
ITPs may not be bug-free and may lead to errors in generated proofs while not being apparent within the proof systems themselves. Together with possible mistakes in the process of importing and exporting, OpenTheory is not guaranteed to be reliable. Even worse, proofs can be huge, making them difficult or even impossible to be checked by hand. The demand of reliability of such systems leads to the necessity of proof checking, especially independently from the ITPs involved. Taking advantage of the similarity of the logic and design between these systems, OpenTheory [9] has developed a standard format for serialising proofs [9]. One way to verify these proofs (also known as proof articles) is to export them to the OpenTheory format followed by the proof checking process by Dedukti [12].

*The author was supported by the MPRI-INRIA scholarship.

2 Proof Transformation and Checking with Holide and Dedukti

Dedukti [12] is a logical framework based on $\lambda\Pi$-calculus Modulo [3] for defining logics and checking proofs. It has been widely used as a universal proof checker for ITPs including Coq, Matita, HOL, FoCaLiZe, etc[1]. To transform HOL proofs from the OpenTheory format to the Dedukti format, we need a translator, namely Holide [1]. Holide uses a modular translation of higher order logic which makes the translation possible to extend [1]. Recent updates of the OpenTheory resulted in an upgrade of the Holide program. More specifically, OpenTheory expanded its logic kernel and added some new inference rules and some other features[2]. Holide is therefore upgraded to version 2 most recently to capture corresponding features and the new format.

3 Proof Retrieval with ProofCloud

ProofCloud[3] is a search engine of higher order proofs customised from Swiftype. While OpenTheory groups proofs up, ProofCloud unpacks them and displays a description of each package (on the index pages) as well as all the theorems contained individually. It has been populated by over 1,800 proofs from 6 packages of OpenTheory. As far as the author knows, it is the only online proof search engine of its kind. ProofCloud presents also the proof checking results by Holide and Dedukti for each package. Further more, it tracks the origin of classicism. With the ability to classify classical proofs, it illustrates the axioms and constructive/classical lemmas involved for each theorem as well as the amount of constructive/classical proofs for the package page.

4 Evaluation and Conclusion

The standard library in OpenTheory grouped theorems into packages, including the standard library and theorems of booleans, sets, lists, etc. Previous work of Holide has checked only the standard library. Following the upgrade of Holide, for the first time, Holide and Dedukti performed proof checking on all packages of OpenTheory. Table 1 illustrates the size of OpenTheory proof article files and the time taken for translation as well as the size of translated files and the time taken for proof checking by Dedukti. Both article files and Dedukti files are compressed by *gzip* to reduce the effect of syntax formatting and white-space. These benchmarks were generated on a 64-bit Intel Core i5-4590 CPU @3.30GHz $\times 4$ machine with 3.8GB RAM. This provides evidence that OpenTheory is a reliable platform for higher order proofs and validated the upgrade of Holide. In addition, the structural proof analyses by ProofCloud shows that the proportion of constructive theorems varies from package to package. For example, the *natural-divides* package has only 10 constructive theorems out of 136 theorems, making only 7.35% proofs constructive in the package. Apart from maintaining Holide, future work also includes adding more packages to ProofCloud and further improve the user interface and the searching accuracy.

[1] `https://www.rocq.inria.fr/deducteam/software.html`

[2] More details between version 5 and version 6 is included in the announcement: `http://www.gilith.com/pipermail/opentheory-users/2014-December/000461.html`

[3] `airobert.github.io/proofcloud/`

Package	Proof Translation with Holide		Proof Checking with Dedukti	
	Size (KB)	Time (s)	Size (KB)	Time (s)
base	1,194	19.42	4,440	9.74
cl	313	5.56	1,219	2.46
empty	0	0.00	0	0.00
gfp	112	1.35	375	0.65
lazy-list	1,391	31.78	5,717	13.11
modular	37	0.37	111	0.17
natural-bits	132	1.39	419	0.68
natural-divides	157	1.94	566	0.99
natural-fibonacci	108	1.24	354	0.60
natural-prime	116	1.34	388	0.65
parser	204	3.15	776	1.69
probability	23	0.23	69	0.11
stream	63	0.73	211	0.38
word10	71	0.62	216	0.29
word12	72	0.75	220	0.35
word16	107	0.77	364	0.36
word5	64	1.56	192	0.72
Total	**4,377**	**72.21**	**15,637**	**32.95**

Table 1: Benchmarks of OpenTheory with Holide and Dedukti

References

[1] Ali Assaf and Guillaume Burel. Translating HOL to dedukti. In *Proceedings Fourth Workshop on Proof eXchange for Theorem Proving, PxTP 2015, Berlin, Germany, August 2-3, 2015.*, pages 74–88, 2015.

[2] Ali Assaf and Raphaël Cauderlier. Mixing HOL and coq in dedukti (extended abstract). In *Proceedings Fourth Workshop on Proof eXchange for Theorem Proving, PxTP 2015, Berlin, Germany, August 2-3, 2015.*, pages 89–96, 2015.

[3] Denis Cousineau and Gilles Dowek. Embedding pure type systems in the lambda-pi-calculus modulo. In *Typed lambda calculi and applications*, pages 102–117. Springer, 2007.

[4] William M Farmer. The seven virtues of simple type theory. *Journal of Applied Logic*, 6(3):267–286, 2008.

[5] Thibault Gauthier and Cezary Kaliszyk. Matching concepts across HOL libraries. In Stephen Watt, James Davenport, Alan Sexton, Petr Sojka, and Josef Urban, editors, *Proc. of the 7th Conference on Intelligent Computer Mathematics (CICM'14)*, volume 8543 of *LNCS*, pages 267–281. Springer Verlag, 2014.

[6] Thibault Gauthier and Cezary Kaliszyk. Matching concepts across hol libraries. In *Intelligent Computer Mathematics*, pages 267–281. Springer, 2014.

[7] Thibault Gauthier and Cezary Kaliszyk. Sharing HOL4 and HOL Light proof knowledge. In *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'15)*, LNCS, 2015. to appear.

[8] John Harrison. Hol light: An overview. In *Theorem Proving in Higher Order Logics*, pages 60–66. Springer, 2009.

[9] Joe Hurd. The OpenTheory standard theory library. pages 177–191.

[10] Cezary Kaliszyk and Alexander Krauss. Scalable lcf-style proof translation. In *Interactive Theorem Proving*, pages 51–66. Springer, 2013.

[11] Marcel Oliveira, Ana Cavalcanti, and Jim Woodcock. Unifying theories in proofpower-z. In *Unifying Theories of Programming*, pages 123–140. Springer, 2006.

[12] Ronan Saillard. Dedukti: a universal proof checker. In *Foundation of Mathematics for Computer-Aided Formalization Workshop*, 2013.

[13] Konrad Slind and Michael Norrish. A brief overview of hol4. In *Theorem Proving in Higher Order Logics*, pages 28–32. Springer, 2008.

[14] Makarius Wenzel. Interactive theorem provers from the perspective of isabelle/isar.