Towards the Integration of an Intuitionistic First-Order Prover into Coq

Fabian Kunze
Saarland University, Germany
s9fakunz@stud.uni-saarland.de

Abstract

An efficient intuitionistic first-order prover integrated into Coq is useful to replay proofs found by external automated theorem provers. We propose a two-phase approach: An intuitionistic prover generates a certificate based on the matrix characterization of intuitionistic first-order logic; the certificate is then translated into a sequent-style proof. By additionally preprocessing the formula and postprocessing the proof, parts of the richer, higher-order type theory of Coq can be encoded.

1 Introduction

Sledgehammer [1] and HOLyHammer [2] drastically improved the productivity for users of proof assistants. They make the capabilities of automated theorem provers (ATPs) available from within interactive proof assistants.

The large, monolithic design of state-of-the-art theorem provers can not be easily trusted to be free of bugs. Thus invoking theorem provers as an oracle is unacceptable for most users. Proof assistants are more trustworthy because all reasoning is checked by a kernel intentionally kept small.

To integrate external provers, small yet efficient, *certified* provers *integrated* into the proof assistant are used: Although it is often possible to mechanically translate the proof to a format accepted by the proof assistant, the integrated prover allows for the reconstruction without the full knowledge of all axioms and rules used by the external prover. Thus an integrated prover simplifies the integration of not only one but different external provers.

There has been effort to integrate classical provers into Coq, e.g. SMTCoq [3], Satallax [4] and why3 [5], but they produce proofs that assume classical axioms. As a fair amount of proof developments avoids assuming additional axioms, the acceptance of a future 'Coq Hammer' benefits from the integration of an efficient, *intuitionistic* prover.

2 Existing Intuitionistic Provers in Coq

The existing intuitionistic first-order provers integrated into Coq are not very strong. We evaluated `firstorder` [6], a built-in tactic based on a sequent calculus, and JProver [7], a plugin available for Coq. We considered first-order problems that are likely to emerge in a future 'Coq Hammer'.

For example, we tested formulas where the instantiate of quantifiers is not immediately determined using a goal-driven approach:

$$\begin{aligned}&(\forall x, x = x) \wedge (\forall x, Px \vee Qx)\\ \wedge\ &(\forall xy, x = y \wedge Px \Rightarrow Ry) \wedge (\forall xy, x = y \wedge Qx \Rightarrow Ry) \Rightarrow (\forall x, Rx).\end{aligned}$$

On this formula, `firstorder` was unable to find a proof, even after running multiple minutes. JProver suceeded in less than one second.

We also invoked both provers on several set-theoretical problems from the ILTP (Intuitionistic Logic Theorem Proving) library [8]. We filtered unnecessary axioms of set theory, resulting in problems like

$$\begin{aligned}&(\forall ABX, X \in A \cup B \Leftrightarrow X \in A \vee X \in B)\\ \wedge\ &(\forall AB, A = B \Leftrightarrow A \subset B \wedge B \subset A)\\ \wedge\ &(\forall AB, A \subset B \Leftrightarrow \forall X, X \in A \Rightarrow X \in B) \Rightarrow (\forall A, A \cup A = A).\end{aligned}$$

On this and similar problems, both `firstorder` and JProver failed to find proofs after running several minutes. Therefore, faster intuitionistic provers integrated into Coq are necessary for a 'Coq Hammer' used in practice.

3 Proposed Architecture

We propose to employ the recent improvements on automated, intuitionistic first-order theorem proving by Otten: ileanCoP [9, 10] and the forthcoming intuitionistic version of nanoCoP [11, 12]. Both are based on the existence of proof certificates for the matrix characterization of (intuitionistic) validity [13], which can be translated to sequent-style proofs [14].

This architecture is similar to that of JProver, but uses a more efficient proof search procedure, leading to a higher success rate.

3.1 Finding Proof Certificates

The performance of ileanCoP is well in identifying true formulas compared to other intuitionistic provers [10], but it does not keep track of the proof found. Furthermore, it is based on a *clausal* variant of the matrix characterization for intuitionistic logic. The necessary translation into a non-clausal matrix proof has been sketched in the correctness proof of ileanCoP [9], but to our knowledge has not yet been implemented.

The classical prover nanoCoP [11] solves both problems: It outputs the proof certificate found and uses the non-clausal matrix characterization of classical validity. Otten is currently extending nanoCoP to an intuitionistic variant by integrating prefix unification [13], a method already employed to derive ileanCoP from the classical prover leanCoP.

In our proposed architecture, the proof certificate for a first-order formula consists of a pair of substitutions $\sigma = (\sigma_Q, \sigma_J)$ and a set of pairs of σ-complementary literals in the formula, called connections. Two literals are σ-complementary if they are complementary under the term substitution σ_Q and their positions in the formula are compatible with σ_J. Thus σ_Q resolves the usual need for non-circular instantiation, while σ_J ensures that intuitionistic restrictions to the order in which one decompose different parts of the formula are satisfied. These order restrictions are one of the main difficulties in intuitionistic first-order theorem proving.

3.2 Generating Sequence Proofs

The translation of a matrix characterisation proof certificate into a sequent-style proof has already been investigated and implemented for JProver[14]. We intend to adopt this translation.

4 Discussion

Modular vs Monolithic

We explicitly want to use a modular implementations for the two phases, possibly written in multiple languages. The Prolog version of the intuitionistic variant of nanoCoP is expected to materialize soon and there already is an implementation of the sequence proof generating algorithm integrated into Coq. Thus we expect no challenge in creating a prototype of the suggested architecture using the Prolog program. This would allow to test whether proposed setup is suitable for the intended use case.

In the longer term, it would be desirable to have a native OCaml implementation of the proof search procedure, allowing for a deployment within Coq, without additional binaries. The classical leanCoP has been ported to OCaml for the HOL light proof assistant, with performence comparable to the Prolog version[15]. This port can serve as a starting point for a native OCaml version of the forthcoming intuitionistic nanoCoP. Then, the modular approach allows to optionally use external proof procedures. This allows to evaluate improvements to the Prolog proof procedure before porting them.

Explicit Proofs vs Reflection

One approach in automation in Coq is 'proof by reflection': The proof search procedure is both written and certified in Coq.

We propose to generate native Coq proofs instead: This allows for transformations on the formula and the proof, to incorporate aspects of the higher order type theory, e.g. annotations with predicates containing type information.

Intuitionistic vs Classical

Automated theorem proving in intuitionistic logic is computationally harder than in classical logic. For developments assuming classical axioms, the intuitionistic part of both phases can be made optional, resembling the classical proof search of nanoCoP without significant overhead.

Acknowledgements

We thank Jens Otten for his helpful discussions and suggestions, and Jasmin Blanchette for his comments on this extended abstract.

References

[1] Lawrence C. Paulson and Jasmin Christian Blanchette. Three years of experience with Sledgehammer, a practical link between automatic and interactive theorem provers. In Geoff Sutcliffe, Stephan Schulz, and Eugenia Ternovska, editors, *The 8th International Workshop on the Implementation of Logics, IWIL 2010, Yogyakarta, Indonesia, October 9, 2011*, volume 2 of *EPiC Series*, pages 1–11. EasyChair, 2010.

[2] Cezary Kaliszyk and Josef Urban. HOL(y)Hammer: Online ATP service for HOL Light. *Mathematics in Computer Science*, 9(1):5–22, 2015.

[3] Michaël Armand, Germain Faure, Benjamin Grégoire, Chantal Keller, Laurent Théry, and Benjamin Werner. A modular integration of SAT/SMT solvers to Coq through proof witnesses. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs—First International Conference, CPP 2011, Kenting, Taiwan, December 7-9, 2011. Proceedings*, volume 7086 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2011.

[4] Chad E. Brown. Satallax: An automatic higher-order prover. In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *Automated Reasoning—6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, volume 7364 of *Lecture Notes in Computer Science*, pages 111–117. Springer, 2012.

[5] François Bobot, Jean-Christophe Filliâtre, Claude Marché, and Andrei Paskevich. Why3: Shepherd your herd of provers. In K. Rustan M. Leino and Michał Moskal, editors, *Boogie 2011*, pages 53–64, 2011.

[6] Pierre Corbineau. *First-Order Reasoning in the Calculus of Inductive Constructions*, pages 162–177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[7] Stephan Schmitt, Lori Lorigo, Christoph Kreitz, and Alexey Nogin. JProver : Integrating connection-based theorem proving into interactive proof assistants. In R. Gore, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 421–426. Springer Verlag, 2001.

[8] Thomas Raths, Jens Otten, and Christoph Kreitz. The ILTP problem library for intuitionistic logic: Release v1.1. *Journal of Automated Reasoning*, 38(1-3):261–271, April 2007.

[9] Jens Otten. Clausal connection-based theorem proving in intuitionistic first-order logic. In Bernhard Beckert, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: 14th International Conference, TABLEAUX 2005, Koblenz, Germany, September 14-17, 2005. Proceedings*, pages 245–261, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[10] Jens Otten. leancop 2.0 and ileancop 1.2: High performance lean theorem proving in classical and intuitionistic logic (system descriptions). In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Automated Reasoning: 4th International Joint Conference, IJCAR 2008 Sydney, Australia, August 12-15, 2008 Proceedings*, pages 283–291, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[11] Jens Otten. nanocop: A non-clausal connection prover. In *Automated Reasoning: 8th International Joint Conference, IJCAR 2016 Coimbra, Portugal, June 27 -July 2, 2016 Proceedings*, 2016. to appear.

[12] Jens Otten. personal communication, 2016.

[13] Lincoln Wallen. *Automated Deduction in Nonclassical Logics*. MIT Press, Cambridge, MA, USA, 1990.

[14] Stephan Schmitt and Christoph Kreitz. Converting non-classical matrix proofs into sequent-style systems. In *Automated Deduction — Cade-13: 13th International Conference on Automated Deduction New Brunswick, NJ, USA, July 30 – August 3, 1996 Proceedings*, pages 418–432, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[15] Cezary Kaliszyk, Josef Urban, and Jiři Vyskočil. Certified connection tableaux proofs for hol light and tptp. In *Proceedings of the 2015 Conference on Certified Programs and Proofs*, CPP '15, pages 59–66, New York, NY, USA, 2015. ACM.