24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

The Higher-Order Prover Leo-III

Alexander Steen[1] and **Christoph Benzmüller**[2]

Abstract. Leo-III is an effective automated theorem prover for extensional type theory with Henkin semantics. It is based on an extensional higher-order paramodulation calculus and supports reasoning in monomorphic and rank-1 polymorphic first-order and higher-order logics. Leo-III also automates various non-classical logics, including almost every normal higher-order modal logic.

1 Introduction

Leo-III is an automated theorem prover (ATP) for extensional type theory (also referred to as *classical higher-order logic*, HOL [2]) with Henkin semantics and choice. The system is implemented in Scala, open-source and freely available under a BSD license.[3] It is the successor of the LEO-II prover [3], whose development significantly influenced the build-up of the TPTP THF infrastructure [13] for reasoning in full higher-order logic.

The logical formalisms supported by Leo-III include HOL as its primary target language, but also first-order and propositional logics. As input formats, Leo-III supports all common TPTP [12, 13] dialects (CNF, FOF, TFF, THF) as well as the polymorphic variants TF1 and TH1 [6, 9]. It is one of the few stand-alone ATP systems for polymorphic higher-order reasoning to date. The prover returns results according to the standardized SZS ontology and additionally produces a TSTP-compatible proof certificate [12], if a proof is found. Furthermore, Leo-III natively supports reasoning in almost every normal higher-order (HO) modal logic, including – but not limited to – logics **K**, **D**, **T**, **S4** and **S5** with constant, cumulative or varying domain quantifiers and both global and local notions of consequence [5], and multi-modal combinations thereof.

Multiple evaluation studies underline the practical contribution of the Leo-III prover to the field: the first author's thesis presents an extensive evaluation on different benchmarks sets, including monomorphic and polymorphic HOL problems as well as modal logic problems, which demonstrates its effectiveness in different application areas [10, §6]. A large independent evaluation study of 19 different first- and higher-order ATP systems, called GRUNGE [7], suggests that Leo-III is the most effective reasoning system (number of successfully solved problems) and also the most versatile to date (in terms of supported logical formalisms). Additionally, Leo-III won the LTB division of the 2019 edition of the CADE ATP System Competition (CASC) that is of particular relevance for practical applications of ATP systems.[4]

Full details on Leo-III, its underlying theory, architecture and implementation are presented elsewhere [2, 10, 11].

[1] University of Luxembourg, Luxembourg, email: alexander.steen@uni.lu
[2] Freie Universität Berlin, Germany, email: c.benzmueller@fu-berlin.de
[3] See the Leo-III project at GitHub: github.com/leoprover/Leo-III.
[4] See tptp.org/CASC/27 for details on the competition and its results.

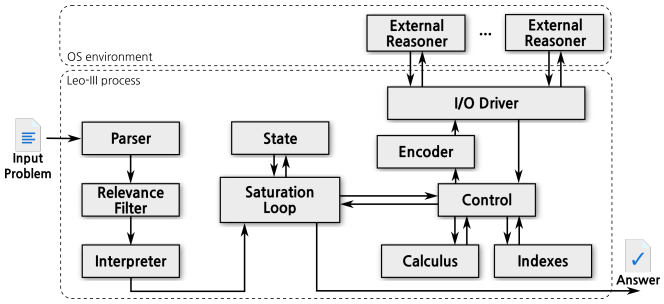

Figure 1. Schematic diagram of Leo-III's architecture. The arrows indicate directed information flow. The external reasoners are executed asynchronously (non-blocking) as dedicated processes of the operating system.

2 Calculus and Implementation

Leo-III is a refutational reasoning system. The initial, possibly empty, set of axioms and the negated conjecture are transformed into an equisatisfiable set of formulas in clause normal form, which is then iteratively saturated until the empty clause is found.

Leo-III extends the paramodulation calculus EP [10] with practically motivated, partly heuristic inference rules. EP consist of generating inferences (such as paramodulation, equality factoring and primitive substitution [1]), inference rules for extensionality treatment, and rules for clausification and unification. Unification constraints are encoded as negative equality literals in the result clause and are solved eagerly by a Huet-style unification procedure. Further calculus rules within Leo-III implement, among others, function synthesis, equational simplificaton routines, and special treatment of injective function symbols. A detailed description of EP and its extensions can be found in the first author's PhD thesis [10].

Theorem 1 (Soundness and Completeness [10]) *The EP calculus is sound and refutationally complete for HOL with Henkin semantics.*

Fig. 1 displays the top-level architecture of Leo-III. A control layer selects heuristically which inferences rules are applied in which parameter setting during saturation, handles indexing data structures, and bridges to the driver that connects to external reasoning systems. Leo-III collaborates with such external systems, in particular, with first-order ATPs such as E, iProver and Vampire as well as SMT solvers, e.g. with CVC4. Cooperation is not restricted to first-order systems, and further specialized systems such as higher-order model finders may be utilized by Leo-III. Additionally, Leo-III uses several heuristics to restrict the number of inferences, including a higher-order term ordering and a depth-bounded unification procedure. While these restrictions sacrifice completeness in general, evaluations nevertheless confirm practicality of this approach.

```
thf(conj,conjecture,
    ~ ( ? [H:($i>$o)>$i]:
        ! [P:$i>$o, Q:$i>$o]:
          ( ((H @ P) = (H @ Q))
             => (P = Q) ) )).
```

Figure 2. Injective Cantor Theorem in TPTP THF syntax. Universal and existential quantification are written as `!` and `?`, respectively, followed by a list of bound variables and their types. In THF, conjunction, disjunction, negation and implication are written as `&`, `|`, `~` and `=>`, respectively. Function application is denoted with an explicit `@` operator.

3 Application Examples

Logical and mathematical reasoning is one of the main applications of HOL ATP systems, in particular, reasoning about sets, functions and relations is natively supported by its underlying λ-calculus without the need of cumbersome and less effective set axiomatizations as required in first-order logic. Figure 2 displays a formulation of Cantor's theorem based on injective functions in the machine-readable TPTP THF syntax standard for HOL ATP systems. The conjecture, i.e. the non-existence of an injective function from some power set in its underlying set, can be proven using Leo-III in approx. $1.5\,\mathrm{s}$ on a standard computer, and a verifiable proof certificate is generated. The generated proof thereby employs a non-trivial diagonalization argument, involving a generated left-inverse of the supposedly injective function, which is automatically synthesized by Leo-III. This problem was not solved by any other ATP system before.

The expressivity of higher-order logic has recently been utilized for encoding various expressive non-classical logics within HOL. Semantical embeddings of, among others, higher-order modal logics, conditional logics, many-valued logics, deontic logics, free logics and combinations thereof can be used to automate reasoning within the respective logic using ATP systems for classical HOL. A prominent result from the applications of automated reasoning in non-classical logics, here in quantified modal logics, is the computer-assisted detection of an inconsistency in Gödel's Ontological Argument [4] that was unknown for years. The semantical embedding approach as means for automation of quantified modal logics has been integrated into the Leo-III prover, turning it into an effective ATP system for many HO quantified normal (multi-)modal logics [8]. Figure 3 displays an example modal logic statement that can be proved by Leo-III in under $300\,\mathrm{ms}$. Up to the authors' knowledge, no other automated reasoning system currently supports native HO modal logic reasoning, in particular with flexible semantical parameters that are specified by the user.

4 Summary

Leo-III is a state-of-the-art reasoning system for full higher-order logic, offering many relevant features and capabilities. Due to its wide range of natively supported classical and non-classical logics, which include polymorphic higher-order logic and numerous first-order and higher-order modal logics, the system has many topical applications in computer science, AI, maths and philosophy.

Several evaluations on heterogeneous benchmark sets show that Leo-III is one of the most effective HO ATP systems to date, and it also plays a pivotal role in the ongoing extension of the TPTP library and infrastructure to support modal logic reasoning. A long term goal of the Leo-III prover is to provide automated reasoning for a wide range of non-classical logics and their combinations.

```
thf(spec, logic, ( $modal := [
  $constants := $rigid,
  $quantification := $constant,
  $consequence := $global,
  $modalities := $modal_system_S5] )).

thf(becker, conjecture,
  ! [P:$i>$o, F:$i>$i, X:$i]: (? [G:$i>$i]:
      ( ($dia @ ($box @ (P @ (F @ X))))
         => ($box @ (P @ (G @ X))) ) )).
```

Figure 3. A corollary of Becker's postulate, given by the formula $\forall P_{\iota\to o}.\forall F_{\iota\to\iota}.\forall X_\iota.\exists G_{\iota\to\iota}.(\Diamond\Box P(F(X)) \Rightarrow \Box P(G(X)))$. The first five lines specify the modal logic (a S5 logic with rigid constants, constant domains and global consequence) under which the problem is to be analyzed. The modal operators $\Box$ and $\Diamond$ are represented by `$box` and `$dia`, respectively.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable feedback. The work was supported by the German National Research Foundation (DFG) under grant BE 2501/11-1 (Leo-III) and the Volkswagen Foundation (project CRAP).

REFERENCES

[1] Christoph Benzmüller, 'Higher-order automated theorem provers', in *All about Proofs, Proof for All*, eds., David Delahaye and Bruno Woltzenlogel Paleo, Mathematical Logic and Foundations, 171–214, College Publications, London, UK, (2015).
[2] Christoph Benzmüller and Peter Andrews, 'Church's type theory', in *The Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, Metaphysics Research Lab, Stanford University, summer 2019 edn., (2019).
[3] Christoph Benzmüller et al., 'The higher-order prover LEO-II', *J. Autom. Reasoning*, **55**(4), 389–404, (2015).
[4] Christoph Benzmüller and Bruno Woltzenlogel Paleo, 'Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers', in *ECAI*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 93–98. IOS Press, (2014).
[5] Patrick Blackburn, Johan van Benthem, and Frank Wolter, *Handbook of modal logic*, volume 3, Elsevier, 2006.
[6] Jasmin C. Blanchette and A. Paskevich, 'TFF1: the TPTP typed first-order form with rank-1 polymorphism', in *CADE*, ed., M. P. Bonacina, volume 7898 of *LNCS*, pp. 414–420. Springer, (2013).
[7] Chad E. Brown et al., 'GRUNGE: A grand unified ATP challenge', in *CADE*, ed., P. Fontaine, volume 11716 of *LNCS*, pp. 123–141. Springer, (2019).
[8] Tobias Gleißner, Alexander Steen, and Christoph Benzmüller, 'Theorem provers for every normal modal logic', in *LPAR*, eds., Thomas Eiter and David Sands, volume 46 of *EPiC Series in Computing*, pp. 14–30. EasyChair, (2017).
[9] Cezary Kaliszyk, Geoff Sutcliffe, and Florian Rabe, 'TH1: the TPTP typed higher-order form with rank-1 polymorphism', in *PAAR*, eds., P. Fontaine, S. Schulz, and J. Urban, volume 1635 of *CEUR Workshop Proceedings*, pp. 41–55. CEUR-WS.org, (2016).
[10] Alexander Steen, *Extensional Paramodulation for Higher-Order Logic and its Effective Implementation Leo-III*, volume 345 of *DISKI*, Akademische Verlagsgesellschaft AKA GmbH, Berlin, 2018. Dissertation, Freie Universität Berlin, Germany.
[11] Alexander Steen and Christoph Benzmüller, 'The higher-order prover Leo-III', in *IJCAR*, eds., Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, volume 10900 of *LNCS*, pp. 108–116. Springer, (2018).
[12] Geoff Sutcliffe, 'The TPTP problem library and associated infrastructure - from CNF to TH0, TPTP v6.4.0', *J. Autom. Reasoning*, **59**(4), 483–502, (2017).
[13] Geoff Sutcliffe and Christoph Benzmüller, 'Automated reasoning in higher-order logic using the TPTP THF infrastructure', *J. Formalized Reasoning*, **3**(1), 1–27, (2010).