Journal of Applied Logic 11 (2013) 91–102

Contents lists available at SciVerse ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

# LEO-II and Satallax on the Sledgehammer test bench

Nik Sultana[a,*], Jasmin Christian Blanchette[b], Lawrence C. Paulson[a]

[a] *Computer Laboratory, University of Cambridge, United Kingdom*
[b] *Institut für Informatik, Technische Universität München, Germany*

A R T I C L E I N F O

*Article history:*
Received 14 August 2012
Received in revised form 21 November 2012
Available online 13 December 2012

*Keywords:*
Higher-order logic
Automatic theorem provers
Proof assistants

A B S T R A C T

Sledgehammer is a tool that harnesses external first-order automatic theorem provers (ATPs) to discharge interactive proof obligations arising in Isabelle/HOL. We extended it with LEO-II and Satallax, the two most prominent higher-order ATPs, improving its performance on higher-order problems. To explore their usefulness, these ATPs are measured against first-order ATPs and built-in Isabelle tactics on a variety of benchmarks from Isabelle and the TPTP library. Sledgehammer provides an ideal test bench for individual features of LEO-II and Satallax, revealing areas for improvements.

## 1. Introduction

Most automatic theorem provers (ATPs) are restricted to first-order formalisms, whereas proof assistants typically support more expressive formalisms such as higher-order logic, type theory, and set theory. Until five years ago, there were only two higher-order ATPs, LEO [6] and TPS [1], both based on classical higher-order logic [18]. Since then, a new generation of higher-order ATPs has emerged: LEO-II by Benzmüller et al. [7,37] and Satallax by Brown et al. [3,17]. This development coincided with the extension of the TPTP (Thousands of Problems for Theorem Provers) infrastructure with a language for encoding problems in higher-order logic (THF0) [8], a collection of benchmark problems [5,36], and a competition category for higher-order provers at CASC [31].

Also in recent years, the world of interactive theorem proving witnessed the development and adoption of Sledgehammer [10,29], a bridge between the proof assistant Isabelle/HOL [28] and several first-order ATPs (including E [33], SPASS [38], Vampire [32], and Z3 [19]). When invoked on a proof goal, Sledgehammer heuristically selects a few hundred background facts, translates them to first-order logic, invokes the external provers in parallel, and reconstructs the proofs in Isabelle. Sledgehammer performs remarkably well in empirical evaluations [10,14] and boosts user productivity [22]. But Paulson has remarked [29, §4],

> Sledgehammer's performance on higher-order problems is unimpressive, and given the inherent difficulty of performing higher-order reasoning using first-order theorem provers, the way forward is to integrate Sledgehammer with an actual higher-order theorem prover.

This paper presents a double materialisation of this vision: an extension of Sledgehammer with LEO-II and Satallax as additional backends (Section 3). The extension reuses many components of Sledgehammer, including the parallel architecture and the relevance filter, but communicates with the ATPs in the higher-order language THF0. Although LEO-II, Satallax, and Isabelle all support "higher-order logic", the translation from Isabelle to THF0 is nontrivial because THF0 does not cater for polymorphic types and axiomatic type classes, which are ubiquitous in Isabelle formalisations.

* Corresponding author.
*E-mail addresses:* ns441@cam.ac.uk (N. Sultana), blanchette@in.tum.de (J.C. Blanchette), lp15@cam.ac.uk (L.C. Paulson).

1570-8683/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jal.2012.12.002

The integration is useful for proving goals where higher-order features predominate, as demonstrated by a few examples (Section 4). To ascertain more precisely the potential of LEO-II and Satallax, we let them compete on standard Isabelle benchmarks against first-order ATPs and built-in Isabelle tactics (Section 5.1). Although they are nowhere as powerful as the first-order ATPs, they can occasionally solve problems that no other provers or tactics can solve. To make the evaluation more informative, the Isabelle problems are complemented by a subset of the TPTP library, which emphasises the higher-order aspects of the logic. By tuning Sledgehammer's translation, we carried out a fine-grained evaluation (Section 5.2) of the higher-order ATPs' handling of types and $\lambda$-abstractions (two problem features we would expect them to handle well) and large background theories. Sledgehammer then acts as a test bench for LEO-II and Satallax, suggesting avenues for improvements.

## 2. Background

This paper combines several technologies—TPTP, LEO-II, Satallax, Isabelle/HOL, and Sledgehammer—that are amply described elsewhere. This section briefly outlines them.

### 2.1. TPTP formats

The TPTP infrastructure defines a hierarchy of languages [8,35]. Of interest to us are the *first-order form* (FOF) for first-order logic with equality over untyped terms, the core *typed first-order form* (TFF0) that extends FOF with simple types (sorts), and the core *typed higher-order form* (THF0) for higher-order logic. Ignoring minor syntactic differences, the strict inclusions FOF $\subset$ TFF0 $\subset$ THF0 hold.

THF0 types are either *type constants* $\kappa$ or the function type $\sigma \to \tau$, where $\sigma$ and $\tau$ are arbitrary types. The types of propositions $o$ and of individuals $\iota$ are predefined. The intended semantics of THF0 is Henkin semantics with extensionality and Hilbert choice. We take some liberties with the syntax, preferring traditional notations and omitting the apply operator @; thus, we write $\mathsf{f}\ X\ Y$ rather than $\mathsf{f} @ X @ Y$ for the application of $X$ and $Y$ to the curried function f. We do honour the TPTP convention that variable names start with an uppercase letter and constants with lowercase. The use of sans serif for constants further emphasises this distinction.

### 2.2. LEO-II and Satallax

The higher-order automatic provers LEO-II [7] and Satallax [3,16] have THF0 as their input language. Both attempt to find a refutation from the negated conjecture and the axioms, amounting to a proof of the original conjecture. To improve their effectiveness, both provers implement strategy scheduling, which involves trying a sequence of option settings, each for a fraction of the allotted time.

LEO-II implements a higher-order resolution calculus and periodically dispatches first-order subproblems to a first-order prover, usually E, with which it cooperates. LEO-II features several optimisations, notably shared indexed terms [37] and a simple relevance filter that is activated for problems that contain 100 axioms or more. Its proofs, expressed in the TSTP format, combine native inferences with embedded E proofs [34].

Satallax is a tableau-based instantiation prover. It builds propositional approximations of a THF0 problem and relies on the SAT solver MiniSat [20] to check them. In case of success, Satallax can return an unsatisfiable core (a list of formulas that suffice to obtain a contradiction), a Coq proof script, or a Coq proof term.

### 2.3. Isabelle/HOL

Isabelle [28] is a logical framework that provides a metalogic to encode the semantics of object logics and a collection of basic definitions to manage inference in those logics. Isabelle follows the LCF architecture: the logical kernel defines an abstract datatype of theorems, and theorems are proved using only the methods made available by the kernel.

HOL is Isabelle's most developed object logic. It is based on classical higher-order logic (simple type theory) [18], augmented with Hilbert choice, polymorphism, and Haskell-style axiomatic type classes [21,40]. Users of Isabelle/HOL invariably work within a substantial body of formalised mathematics that has been constructed on the foundation of pure higher-order logic, including the concepts of orders, lattices, sets, functions, relations, numbers, and lists. We adhere to the Isabelle/HOL conventions for writing terms and rely on the reader's discernment to identify $x$ in Isabelle with $X$ in THF0, 0 with zero, () with unity, Cons with cons, and so on.

Isabelle/HOL includes several automatic proof tools, including a term-rewriting engine (the *simplifier*), a tableau prover, and decision procedures for specific theories. It also works with external provers through Sledgehammer.

### 2.4. Sledgehammer

The purpose of Sledgehammer is to prove theorems with no user effort using automatic theorem provers—historically, first-order resolution provers and SMT solvers [10,25]. When activated (by a single mouse click), it packages up the formula to be proved along with a collection of relevant facts (definitions, lemmas, or axioms) extracted from Isabelle's libraries