Hybrid and alternative logics in Isabelle

Joshua Chen

Computational Logic
Department of Computer Science
University of Innsbruck

0 Introduction

I'm interested in making proof assistants more powerful and at the same time simpler to use, so as to become attractive as regular tools for mathematicians in particular. Additionally, I would like to see the formal proof community achieve greater interoperability across the different proof assistants and libraries that exist today. To this end, I work on the following topics.

1 Soft type infrastructure for Isabelle

The soft type paradigm [Kra10; LP99] is an approach that views types in terms of their defining logical predicates, as opposed to the type-theoretic approach in which types are primary and logic follows after via the Curry-Howard isomorphism. In the soft type framework, a type T is defined by a predicate ϕ_T, and saying that a term t is of type T simply means that $\phi_T(t)$. This viewpoint corresponds more closely to the way mathematicians think about and use types, and provides a powerful extension to untyped formalisms, as well as formalisms with weaker type systems.

Together with my advisor and collaborators I am working on developing soft type infrastructure for Isabelle. Among other things, this involves writing code to automate type inference, inference of implicit arguments and the derivation of additional type information. For the moment we focus on Isabelle/HOL, but also work as generally as possible in order to be able to support Isabelle/Pure later. Discussion is also currently ongoing with the developers of Isabelle in order to see how much soft type support we can incorporate into its native functionality.

2 Set theory as a foundation for interactive proof

Softly-typed higher-order Tarski-Grothendieck

I am also working on a softly-typed object logic for Isabelle built on top of Isabelle/HOL and based on higher-order Tarski-Grothendieck (HOTG) set theory. We aim to use the experience gained in developing Isabelle/Mizar [KP18] in order to build a simplified softly-typed set-theoretic foundation that encompasses the Mizar mathematical library (MML) alongside Isabelle/ZF and potentially other set-theoretic libraries.

The Isabelle/ZF library should be supported almost out of the box, since its first-order logical foundations are contained in HOL. It will only remain to prove the axioms of ZF from those of HOTG, as well as provide a compatibility layer in order to use definitions and constructions particular to Isabelle/ZF.

Porting the MML into the proposed simplified foundation will require more work. Mizar's semantics are quite complicated, and its automation is fine-tuned to a degree that allows for very succinct proofs. A successful port would translate the Mizar (re-)definitions of modes, adjectives, predicates, and functors into soft types, HOL predicates, and function or term definitions in HOTG. In addition, we would need to write methods and other

In: C. Kaliszyk, E. Brady, J. Davenport, W.M. Farmer, A. Kohlhase, M. Kohlhase, D. Müller, K. Pąk, and C. Sacerdoti Coen (eds.): Joint Proceedings of the FMM and LML Workshops, Doctoral Program and Work in Progress at the Conference on Intelligent Computer Mathematics 2019 co-located with the 12th Conference on Intelligent Computer Mathematics (CICM 2019), Prague, Czech Republic, July 8–12, 2019, published at http://ceur-ws.org

proof automation to be able to maintain the structure of the MML as much as possible (which is a requirement if we are to have any hope of automating the porting process).

Structural set theory

Structural set theory is an interesting potential alternative to traditional set theory and type theory as a foundation for formal proof. It differs from traditional "material" set theory in its lack of a global membership relation—in particular, the elements of sets are themselves not sets in general. Structural set theories mix aspects of traditional untyped set theory together with type theory, and it would be interesting to see what, if any, advantages such a foundation would provide in practical formalization work. I have started an Isabelle formalization of Michael Shulman's *Sets, Elements, and Relations* [nLa19] in order to investigate this, and will experiment with writing some basic mathematics as time permits.

3 Dependent types for HOL

Initial work with implementing homotopy type theory as an Isabelle object logic [Che19] has indicated the potential of the soft type approach to adding dependent types on top of simple type theory [JM93].

Together with current work on soft type infrastructure for Isabelle, it would be interesting to see how much of the practical functionality of dependently-typed proof assistants can be implemented in a HOL-based system. This will involve specializing the soft type inference and implicit argument inference tools to type theoretic object logics based on Isabelle/Pure, and also writing methods that emulate the tactic-based proof functionality of Coq, Lean, and other dependently-typed systems.

A major technical issue to overcome in such a work is the handling of the derivation of proof terms. At the moment, the only way to do this in Isabelle is by using the `schematic_goal` command, which is too restrictive to allow many basic operations common in such derivations. In particular, induction inside a schematic goal context creates difficulties with Skolem terms. It is also not currently possible to begin "sub-schematic goals" inside an existing schematic goal. Determining if these difficulties are resolvable—and if so, how one would go about such a task—is a longer-term research goal.

References

[Che19] Joshua Chen. *Isabelle/HoTT*. GitHub repository. 2019. URL: `https://github.com/jaycech3n/Isabelle-HoTT`.

[JM93] Bart Jacobs and Tom Melham. "Translating Dependent Type Theory into Higher Order Logic". In: *Typed Lambda Calculi and Applications*. Ed. by Marc Bezem and Jan Friso Groote. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 209–229. ISBN: 978-3-540-47586-6.

[KP18] Cezary Kaliszyk and Karol Pąk. "Semantics of Mizar as an Isabelle Object Logic". In: *Journal of Automated Reasoning* (Aug. 2018). ISSN: 1573-0670. DOI: `10.1007/s10817-018-9479-z`. URL: `https://doi.org/10.1007/s10817-018-9479-z`.

[Kra10] Alexander Krauss. *Adding Soft Types to Isabelle*. `https://www21.in.tum.de/~krauss/papers/soft-types-notes.pdf`. Unpublished note. 2010.

[LP99] Leslie Lamport and Lawrence C. Paulson. "Should Your Specification Language Be Typed?" In: *ACM Trans. Program. Lang. Syst.* 21.3 (May 1999), pp. 502–526. ISSN: 0164-0925. DOI: `10.1145/319301.319317`. URL: `http://doi.acm.org/10.1145/319301.319317`.

[nLa19] nLab authors. *SEAR*. `http://ncatlab.org/nlab/show/SEAR`. Revision 54. May 2019.