PhD Proposal: Polarized Deduction Modulo Theory

Olivier Hermant
CRI, MINES ParisTech

May 15, 2017

1 Formal Methods for Safer Software

Software industry is building up the most complex artificial objects ever seen, from several orders of magnitude. With such complexity come misfunctionments (bugs) and vulnerabilities of those systems, that one can ultimately trace back to a misconception or a faulty implementation.

Formal methods help increasing quality of software, in particular in life-critical domains. Several techniques have been investigated and are now employed at an industrial scale, like model checking, abstract interpretation, code analysis, proof assistants and automated theorem proving. They have seen a growing interest in the past decades.

This PhD thesis proposal is at the crossroad of proof assistants and automated theorem proving, and at the same time at the crossroad of implementation and proof theory. It aims at developing two tools, by enhancing them with Polarized Rewriting: Dedukti (proof checker) and Zenon Modulo (automated theorem prover) as well as the theoretical properties of the associated logical frameworks. Those tools, despite their young age, have already been used in an industrial platform for safe-by-construction software [The12].

2 The Framework: Deduction Modulo Theory

The gist of Deduction Modulo Theory is to embed computation within proof systems, by the means of *rewrite rules*. This speeds up theorem provers by avoiding the need for axioms and emphasizing the computational and deterministic nature of parts of proofs. It also offers a versatile and efficient way to express proof assistants in a shallow way, when combine with a type system like LF.

Polarized Deduction Modulo Theory [Dow10] is an improvement over Deduction Modulo Theory [DHK03], that allows rewrite rules to be selectively applied to the hypothesis or to conclusion side of proofs.

Currently, it has been successfully implemented in a resolution-based prover [Bur11] and has given very promising results [BBC+17]. One of the avantages of this approach is the possibility to express asymetric axioms. Moreover, the generated rewrite system lends itself well to Skolemization, in particular in classical logic, which further speeds up proof-seach.

3 Research Directions

3.1 Polarized Deduction Modulo Theory in Zenon

The implementation of Polarized Deduction Modulo Theory in a tableau-based theorem prover is one of the objectives of the PhD. The chosen tool is Zenon Modulo [DDG+13], that currently implements *unpolarized* Deduction Modulo Theory.

This implementation will be assessed by an intensive benchmarking, through the TPTP library and the BWare platform [The12]. In particular, a thorough comparison with the pristine version of Zenon Modulo is required.

3.2 Proof Theory of Polarized Deduction Modulo Theory

From the proof-theoretical point of view, nothing has been done so far. The research plan involves the definition of classical and intuitionistic models of Polarized Deduction Modulo Theory, for instance by refining the order relation that can be found in Boolean and Heyting Algebras, but also in generalizing Kripke Structures, etc. Of course, the immediate application is to prove soundness and completeness of the calculus wrt to the semantics, and in a second time to derive cut admissibility theorems by semantic means, in the spirit of [Oka99, BH06b, BH06a, LDM05].

Expressivity of the approach (in link with Sec. 3.1), in particular the possibility to express constraint [LN07] or higher-order systems [LDM05], can also be envisioned.

Lastly, the notion of superconsistency [Dow06] needs an adaptation, probably more profound and more insightful.

3.3 Higher-Order Polarized Deduction Modulo Theory

The polarized rewriting approach can be lifted to *type theory*. The goal here is to introduce polarized rewrite rules in a framework like the $\lambda\Pi$-calculus modulo theory, which currently implements plain rewriting.

The asymetry brought up by polarized rewriting could be used to express subtyping, which is one of the features that are missing to Dedukti currently, and that prevents it to express some systems in a more natural way.

Therefore, besides proof theory, an implementation of polarized rewriting in Dedukti [BCH12] is extremely desirable.

The challenges listed in this section are numerous, and their difficulties range from simple to very difficult.

4 Expectations

It is not expected, that those subjects are covered by a single person. The choice of the more specific area to focus on will depend on the applicant and its preferences.

An M.Sc. specialization in computer science or in mathematics is a strong requirement. A few basic courses either on logics (proof systems, proof assistants), rewrite systems, or on functional programming, are recommended.

Keywords: proofs, rewriting, theorem provers, proof assistants, model theory, first-order logic, type systems

References

[BBC+17] Guillaume Burel, Guillaume Bury, Raphaël Cauderlier, David Delahaye, Pierre Halmagrand, and Olivier Hermant. Automated deduction: When deduction modulo theory meets the practice. 44p. Submitted to Journal of Automated Reasoning., 2017.

[BCH12] Mathieu Boespflug, Quentin Carbonneaux, and Olivier Hermant. The λΠ-calculus modulo as a universal proof language. In *In Second Workshop on Proof Exchange for Theorem Proving (PxTP)*, volume 878, pages 28–43. CEUR-WS.org, 2012.

[BH06a] Richard Bonichon and Olivier Hermant. On constructive cut admissibility in deduction modulo. In Thorsten Altenkirch and Conor McBride, editors, *TYPES for proofs and programs*, volume 4502 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2006.

[BH06b] Richard Bonichon and Olivier Hermant. A semantic completeness proof for tableaux modulo. In Miki Hermann and Andrei Voronkov, editors, *LPAR 2006*, volume 4246 of *Lecture Notes in Computer Science*, pages 167–181, Phom Penh, Cambodia, November 2006. Springer-Verlag.

[Bur11] Guillaume Burel. Experimenting with deduction modulo. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2011.

[DDG+13] David Delahaye, Damien Doligez, Frédéric Gilbert, Pierre Halmagrand, and Olivier Hermant. Zenon modulo: When achilles outruns the tortoise using deduction modulo. In Ken McMillan, Aart Middledorp, and Andreï Voronkov, editors, *LPAR*, volume 8312 of *LNCS ARCoSS*, pages 274–290. Springer, 2013.

[DHK03] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31:33–72, 2003.

[Dow06] Gilles Dowek. Truth values algebras and normalization. In Thorsten Altenkirch and Conor McBride, editors, *TYPES for proofs and programs*, volume 4502 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2006.

[Dow10] Gilles Dowek. Polarized deduction modulo. In *IFIP Theoretical Computer Science*, 2010.

[LDM05] James Lipton and Mary De Marco. Completeness and cut elimination in Church's intuitionistic theory of types. *Journal of Logic and Computation*, 15:821–854, December 2005.

[LN07] James Lipton and Susana Nieva. Higher-order logic programming languages with constraints: A semantics. In Simona Ronchi Della Rocca, editor, *TLCA*, volume 4583 of *Lecture Notes in Computer Science*, pages 272–289. Springer, 2007.

[Oka99] Mitsuhiro Okada. Phase semantic cut-elimination and normalization proofs of first- and higher-order linear logic. *Theoretical Computer Science*, 227:333–396, 1999.

[The12] The BWare Project, 2012. http://bware.lri.fr/.