Outline For FYP: A Solver for Commutative Rings in Agda

Donnacha Oisín Kidney

October 26, 2018

Dependently typed programming languages, such as Agda [4], Idris [1], and Coq [5], extend the type systems of programming languages to allow for proofs to be written as programs. They are of interest to mathematicians and programmers alike: as they're based on a constructivist interpretation of mathematics, the languages themselves are viable avenues for a formal foundation of modern mathematics; as programming languages, many of the features and concepts that have been developed have proven useful in mainstream languages.

Mathematics must be rebuilt from the ground-up in each of these languages. The goal of this project is to build a tool for one particular area in the language Agda.

The tool is an automated solver for equalities in commutative rings. Without the solver, proofs of basic equalities can be hundreds of lines long, and difficult to understand or write. Automating the process isn't as simple as just checking if two sides of an equation are the same, though: the solver has to be *evidence-providing*, meaning that it constructs a formal proof of the equality based on the ring axioms.

Such a solver already exists for Agda [2]: the target implementation for this project is a highly optimized version of the existing solver, which dramatically improves the efficiency. The optimizations are the same as those in Coq's ring solver [3], although the implementation strategy is different, and more properties are verified.

References

[1] E. Brady, "Idris, a general-purpose dependently typed programming language: Design and implementation," *Journal of Functional Programming*, vol. 23, no. 05, pp. 552–593, Sep. 2013. [Online]. Available: http://journals.cambridge.org/article_S095679681300018X

[2] N. A. Danielsson, "The Agda standard library," Jun. 2018. [Online]. Available: https://agda.github.io/agda-stdlib/README.html

[3] B. Grégoire and A. Mahboubi, "Proving Equalities in a Commutative Ring Done Right in Coq," in *Theorem Proving in Higher Order Logics*, ser. Lecture Notes in Computer Science, vol. 3603. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 98–113. [Online]. Available: http://link.springer.com/10.1007/11541868_7

[4] U. Norell and J. Chapman, "Dependently Typed Programming in Agda," p. 41, 2008.

[5] T. C. D. Team, "The Coq Proof Assistant, version 8.8.0," Apr. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1219885