

Automatically And Efficiently Illustrating Polynomial Equalities in Agda—Extended Abstract

Donnacha Oisín Kidney

February 15, 2019

Abstract

We present a new library which automates the construction of equivalence proofs between polynomials over commutative rings and semirings in the programming language Agda [1]. It is asymptotically faster than Agda’s existing solver. We use Agda’s reflection machinery to provide a simple interface to the solver, and demonstrate a novel use of the constructed relations: step-by-step solutions.

The library is available at oisdk.github.io/agda-ring-solver/README.html.

These flaws prompted what’s known as the “foundational crisis of mathematics”: an effort to replace the old foundations with new, sound ones. One sketch of these new foundations was proposed in the form of “Hilbert’s program”.

Hilbert’s program was one such effort. It was too ambitious, however: Gödel’s incompleteness theorems showed that several of the goals were in fact *impossible*.

These days,

Contents

1 Introduction	1
1.1 Formalized and Mechanized Mathematics	1

1 Introduction

1.1 Formalized and Mechanized Mathematics

Mathematics is often thought of as something which builds upwards: adding new work and advances on the foundations of old. However, since the early twentieth century, some have turned their attention in the other direction, examining those foundations and ensuring they are really as solid as they seem. Worryingly, serious flaws were found, Russell’s paradox being perhaps the most famous.

References

- [1] U. Norell and J. Chapman, “Dependently Typed Programming in Agda,” p. 41, 2008.