Paper Plan

Author Names (in Alphabetical Order)

October 25, 2024

Contents

1 Introduction

1.1 Magmas and Equational Laws

Introduce the key definitions, and list some past results. Also, mention OEIS sequences for the number of equations with a given number of operations.

Discuss the state of the art of undecidability, particularly the question: Is the EquationX $\Rightarrow$ Equation Y problem for a single variable undecidable in general?

1.2 Equational Theories Project

Describe the initial aims and history of the project.

2 Results

While a large number of theoretically interesting results are not expected, some notable ones can be listed here with links to blueprints/Lean as necessary. Proofs can be deferred to the

appendix.

- A new short Austin pair: Equation 3944 implies Equation 3588 [**?**], but not for infinite magmas [**?**].

3 Mathematical Foundations

This section covers topics like free magmas (including those relative to theories), a completeness theorem, and confluence (unique simplification).

4 Formal Foundations

Here we describe the Lean framework used to formalize the project, covering technical issues such as:

- Magma operation symbol issues
- Syntax ('LawX') versus semantics ('EquationX')
- "Universe hell" issues
- Additional verification (axiom checking, Leanchecker, etc.)
- Use of the 'conjecture' keyword

4.1 Contributions to Mathlib

None yet, but presumably, some of what we do will be uploadable and should be mentioned.

5 Project Management

Shreyas Srinivas and Pietro Monticone have volunteered to take the lead on this section.

Discuss topics such as:

- Project generation from template

- Github issue management with labels and task management dashboard
- Continuous integration (builds, blueprint compilation, task status transition)
- Pre-push git hooks
- Use of Lean Zulip and polls

5.1 Handling Scaling Issues

Mention early human-managed efforts and the need for forethought in setting up a GitHub organizational structure. Discuss the use of transitive reduction to keep the Lean codebase manageable.

5.2 Other Design Considerations

Explain the meaning of "trusting Lean" in a large project and highlight human issues that may arise, tools for external checks, PR reviews, and good practices like branch protection.

6 Finite Magmas and Other Sources of Counterexamples

Describe various sources of example magmas, including finite and linear magmas, and their role in ruling out implications. Also, discuss the computational and memory efficiencies needed.

7 Metatheorems

List some notable metatheorems, including those that did not mature in time for deployment but may still be useful in the future.

8 Automated Theorem Proving

Describe the automated theorem provers used in the project (Z3, Vampire, egg, etc.) and performance statistics. Explore semi-automated vs. fully automated methods and how these were integrated into the project.

9 AI-assisted Contributions

Current contributions include Claude's assistance with front-end coding, with potential for more as the project progresses.

10 User Interface

Describe visualizations and explorer tools used in the project.

11 Statistics and Experiments

Analyze the implication graph and discuss test sets of implication problems for benchmarking theorem provers. Challenge: How can one automatically assign a difficulty level to an implication?

12 Data Management

Describe how data was handled during the project and how it will be managed going forward.

13 Reflections

Include testimonies from participants and reflections on the project, discussing the balance between automation and human input.

14 Conclusions and Future Directions

Summarize insights and future directions for the project, including potential databases and interesting equational laws.

Acknowledgments

Acknowledgments to the broader Lean Zulip community and smaller contributors not listed as authors.

A Proofs of Theoretical Results

Provide the interesting proofs mentioned in the results section, while routine proofs can refer to the blueprint or Lean.

B Author Contributions

List author contributions, using CRediT categories. Elaborate on how these categories are interpreted and add affiliations and grant acknowledgments.

References