

CPSC-354 Report

Jack Mazac
Chapman University

August 29, 2024

Abstract

This report chronicles the learning journey over the semester in CPSC-354. The course encompassed a diverse set of topics including formal systems, Lean theorem proving, and other key areas in computer science and logic. Throughout the semester, I engaged with a range of concepts, starting with an introduction to formal systems and basic Lean proofs, and gradually progressing to more complex topics. The report documents my notes, homework solutions, and critical reflections on the content covered each week. The goal is to provide a comprehensive overview of my understanding and development in these subjects.

Contents

1	Introduction	1
2	Week by Week	1
2.1	Week 1	1
2.2	Week 2	3
3	Lessons from the Assignments	3
3.1	Precision and Structure in Formal Systems	3
3.2	The Art of Rewriting in Lean	3
4	Conclusion	3

1 Introduction

This report serves as a comprehensive documentation of my learning and development throughout the CPSC-354 course. The content spans from the basics of formal systems and Lean theorem proving to more advanced topics covered later in the semester. The report is structured week by week, with detailed notes, homework solutions, and reflections for each period. The aim is to capture both the technical and conceptual growth experienced over the course of the semester.

2 Week by Week

2.1 Week 1

Notes

In Week 1, I explored the foundational concepts of formal systems through the MU-puzzle and began working with Lean proof tactics. The MU-puzzle introduced me to the idea of rule-based transformations within a

formal system, emphasizing the importance of adhering strictly to the rules—known as the "Requirement of Formality." This concept was mirrored in my Lean exercises, where I learned to apply specific proof tactics to simplify and verify logical statements.

Homework

The Lean tutorial levels 5 through 8 provided practical exercises that reinforced the theoretical concepts from the MU-puzzle. Below is a summary of the steps and lessons learned:

Level 5

In Level 5, I learned how to handle simple arithmetic involving the addition of zero. The steps were as follows:

```
rw [add_zero]
rw [add_zero]
rfl
```

This taught me the importance of simplifying expressions step by step and ensuring that both sides of an equation are identical before applying `rfl`.

Level 6

Level 6 focused on precision rewriting. I applied the following steps:

```
rw [add_zero c]
rw [add_zero b]
rfl
```

This exercise highlighted the need for targeted rewriting to simplify specific parts of an expression while maintaining overall accuracy.

Level 7

In Level 7, I worked with the successor function and addition:

```
rw [succ_eq_add_one]
rfl
```

This reinforced the relationship between successor functions and addition, showing how to simplify such expressions in Lean.

Level 8

Level 8 was the most complex, requiring multiple rewrites to simplify nested successor functions:

```
rw [two_eq_succ_one]
rw [add_succ]
rw [one_eq_succ_zero]
rw [add_succ]
rw [add_zero]
rw [four_eq_succ_three]
rw [three_eq_succ_two]
rw [two_eq_succ_one]
rw [one_eq_succ_zero]
rfl
```

This level emphasized the importance of meticulous, step-by-step simplifications, especially when dealing with Peano arithmetic.

Comments and Questions

How do the abstract rules in formal systems like the MU-puzzle translate into practical applications in software engineering?

2.2 Week 2

...

3 Lessons from the Assignments

Throughout the semester, I encountered various challenges and learning opportunities that contributed to my understanding of formal systems, logic, and Lean theorem proving. Here are the key lessons:

3.1 Precision and Structure in Formal Systems

The early weeks of the course, particularly Week 1, emphasized the precision required when working within formal systems. The MU-puzzle illustrated how even simple rules can lead to complex problem-solving scenarios, mirroring the need for exactness in Lean proofs. As I progressed through the levels in Lean, this precision became even more critical, particularly when dealing with nested functions and arithmetic expressions.

3.2 The Art of Rewriting in Lean

Rewriting is a fundamental tactic in Lean, as demonstrated in Levels 5-8. The ability to identify which parts of an expression to rewrite—and in what order—can make the difference between a successful proof and a failed one. This process is not just mechanical; it involves a deep understanding of the underlying logical structure and the relationships between different components of an expression.

4 Conclusion

This course provided a rigorous exploration of formal systems and Lean theorem proving, both of which are foundational to understanding logic and formal reasoning in computer science.

References

[BLA] Author, [Title](#), Publisher, Year.