# CPSC-354 Report

Connor Jacobs
Chapman University

September 5, 2024

**Abstract**

This report will contain a summary of my learning progress throughout the course, so far including: lean proofs

## Contents

## 1 Introduction

This report will document my learning through the course. The content will be structured week by week, with sections on mathematical notes, homework solutions, and personal reflections on the topics discussed.

## 2 Week by Week

### 2.1 Week 1

#### Mathematical Proof

Proving that $37x + q = 37x + q$ demonstrates the reflexivity property of equality. Reflexivity states that any mathematical expression is equal to itself. In this case, the expression $37x + q$ is compared to itself, and it is immediately clear by the reflexivity of equality that this statement is true.

#### Proof in Lean

In Lean, the theorem $37x + q = 37x + q$ is proven using the ‘rfl‘ tactic. The ‘rfl‘ tactic in Lean stands for "reflexivity" and handles proofs of the form $X = X$. When ‘rfl‘ is executed, Lean verifies that both sides of the equation are equal and the proof is complete.

The command to execute in Lean is simply:

```
rfl
```

### Connection Between Lean and Mathematics

In mathematics, we rely on the axiom of reflexivity to assert that $37x + q = 37x + q$. Likewise, in Lean, the ‘rfl‘ tactic automates this process by invoking the same principle. The use of ‘rfl‘ in Lean serves as a direct representation of reflexivity in mathematical logic, providing an automated and formalized way to conclude that an expression is equal to itself.

Thus, both in Lean and in traditional mathematics, the proof of $37x + q = 37x + q$ is an application of the same fundamental concept: the reflexivity of equality.

### Answers to Levels 5-8

#### Level 5

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

#### Level 6

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

#### Level 7

```
rw [one_eq_succ_zero]
rw [add_succ]
rw [add_zero]
rfl
```

#### Level 8

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

## 2.2 Week 2

### Levels 1-5 Proofs in Lean

#### Level 1

```
induction n with d hd
rw [add_zero]
rfl
```

```
rw [add_succ]
rw [succ_eq_add_one]
rw [succ_eq_add_one]
rw [hd]
rfl
```

### Level 2

```
induction b with a ab
rw [add_zero]
rw [add_zero]
rfl
rw [add_succ]
rw [add_succ]
rw [ab]
rfl
```

### Level 3

```
induction b with d hd
rw [add_zero]
rw [zero_add]
rfl
rw [add_succ]
rw [succ_add]
rw [hd]
rfl
```

### Level 4

```
induction c with d hd
rw [add_zero]
rw [add_zero]
rfl
rw [add_succ]
rw [hd]
rw [add_succ]
rw [add_succ]
rfl
```

### Level 5

```
rw [add_assoc]
rw [add_comm b c]
rw [← add_assoc]
rfl
```

## Mathematical Proofs

**Theorem (Associativity of Addition):** For all natural numbers $a$, $b$, and $c$, the addition operation is associative: $(a + b) + c = a + (b + c)$.

*Proof:* By induction on $c$:

Base case: When $c = 0$, we have:

$$a + b + 0 = a + (b + 0)$$

Using the property of addition with zero, $a + b = a + b$ holds by reflexivity.

Inductive step: Assume the hypothesis holds for $c = d$, i.e., $a+b+d = a+(b+d)$. For $c = \text{succ}(d)$, we have:

$$a + b + \text{succ}(d) = (a + b + d) + \text{succ}(0) = a + (b + d) + \text{succ}(0) = a + (b + \text{succ}(d))$$

Thus, associativity is proven.

**Theorem (Right Commutativity of Addition):** For all natural numbers $a$, $b$, and $c$, we have $(a+b)+c = (a + c) + b$.

*Proof:* Rewrite $(a + b) + c$ using associativity and commutativity:

$$(a + b) + c = a + (b + c) = a + (c + b) = (a + c) + b$$

**Proofs in Lean**

**Proof of `add_assoc` in Lean**

```
induction c with d hd
-- Base case
rw [add_zero]
rw [add_zero]
rfl
-- Inductive step
rw [add_succ]
rw [hd]
rw [add_succ]
rw [add_succ]
rfl
```

**Proof of `add_right_comm` in Lean**

```
rw [add_assoc]
rw [add_comm b c]
rw [← add_assoc]
rfl
```

## 3 Lessons from the Assignments

### 3.1 Key Lessons

The assignments taught the foundational properties in mathematics, such as reflexivity, commutativity, and associativity, and their implementation in Lean. Lean's tactics like 'rfl' and 'rw' simplify proofs by automating logical steps, making it easier to verify mathematical statements.

## 4 Conclusion

I have learned the connection between mathematical proofs and Lean verification.

## References

[BLA] Author, Title, Publisher, Year.