CPSC 354 Report

Andrew Eppich
Chapman University

September 15, 2024

Abstract

Contents

1 Introduction

2 Homework 1

2.1 Question 5

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

2.1.1 Proof Explanation

For this question, the Lean proof is related to the corresponding proof in mathematics because we know that we can use the additive identity property, which says that $x + 0 = x$. By using this, we can simplify $b + 0$ and $c + 0$ easily to get $a + b + c = a + b + c$, which we can determine is the same by the reflexivity property, which states that if $a = b$, then a and b are identical. Therefore, $a + b + c$ is identical to $a + b + c$.

2.2 Question 6

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

2.3 Question 7

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

2.4 Question 8

```
rw [two_eq_succ_one]
rw [one_eq_succ_zero]
rw [add_succ]
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
```

2.5 Discord Question

I was wondering if the computers use of discrete math extends to all program computations or just math computations

3 Homework 2

3.1 Question 1

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

3.2 Question 2

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

3.3 Question 3

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

3.4 Question 4

```
induction a with d hd
rw [zero_add]
rw [zero_add]
rfl
rw [succ_add]
rw [succ_add]
rw [succ_add]
rw [hd]
rfl
```

3.4.1 Explaination

The lean proof relates to the proof in mathematics because it uses induction to solve the problem. Then the Lean proof is solved by solving the equation of the successors. Just like in mathematics it uses simple rules to change the positioning of the parenthesis so each side is exactly the same. This is exactly like how the mathematical proof would be written.

3.5 Question 5

```
induction a with d hd
rw [zero_add]
rw [zero_add]
rw [add_comm]
rfl
rw [add_comm]
```

```
rw [add_comm]
rw [succ_add]
rw [succ_add]
rw [succ_add]
rw [succ_add]
rw [hd]
rfl
```

3.6 Discord Question

I was wondering how discrete math and the recursive algorithms we talked about fit into a programming language and how it actually works

4 Homework 3

4.1 Discord Post

Discord Name: Andrew Eppich. In my literature review with ChatGPT, I explored interpreted vs. compiled programming languages. I found that interpreted languages are changed from user to machine code line by line, which is inefficient. Compiled languages are compiled from user to machine code all at once and then run which makes it faster and easier to spot errors. From there I explored interpreted languages and their role in machine learning as well as their history in machine learning. I first found that compiled was much more efficient than interpreted. I then found out that interpreted is mainly used for machine learning. It is mainly used because of the extensive amount of libraries used with interpreted languages, especially Python. Some of those libraries include NumPy, pandas, scikit-learn, TensorFlow, and Matplotlib. These libraries are crucial for machine learning because they are associated with data processing and deep learning. I then took a look into the history of programming languages with machine learning. I found that at first compiled languages were used from the 1950s-1980s. In 1991, Python was developed which became the standard for machine learning in the early 2000s. Python became the main language for machine learning from 2010 and on because of its libraries TensorFlow and PyTorch. https://github.com/AndrewEppich/LLM-Literature-Review/blob/main/README.md

4.2