

CPSC-354 Report

Chaz Gillette

August 29, 2024

Contents

Week 1: Introduction to Lean and Natural Number Game Tutorial

In week one, we worked with Lean and reviewed our discrete math knowledge in the Number Game Tutorial. Problems and solutions are listed below.

Homework Solutions: Week 1

Level 5

```
a + (b + 0) + (c + 0) = a + b + c.  
  rw [add_zero]  
  rw [add_zero]  
  rfl
```

Level 6

```
a + (b + 0) + (c + 0) = a + b + c.  
  rw [add_zero c]  
  rw [add_zero b]  
  rfl
```

Level 7: succ_eq_add_one Theorem

```
Theorem succ_eq_add_one: For all natural numbers a, we have succ(a)  
= a + 1  
  rw [one_eq_succ_zero]  
  rw [add_succ]  
  rw [add_zero]  
  rfl
```

Level 8: $2 + 2 = 4$

$2 + 2 = 4$.

```
nth_rewrite 2 [two_eq_succ_one] -- only change the second '2' to
'succ 1'.
rw [add_succ]
rw [one_eq_succ_zero]
rw [add_succ, add_zero] -- two rewrites at once
rw [\three_eq_succ_two] -- change 'succ 2' to '3'
rw [\four_eq_succ_three]
rfl
```

Detailed Explanation: Level 7 Proof

I chose to explain the proof for level seven because this is where we make the breakthrough with addition. Our goal is to prove that the successor of a is equal to $a + 1$. So, in step one, we want to rewrite one as the successor of zero. That gives us $\text{succ } n = n + \text{succ } 0$. The next step is 'add_succ' so that $\text{succ } n = \text{succ } (n + 0)$. After that, we can remove the zero by 'rw [add_zero]', which will leave us with $\text{succ } n = \text{succ } n$, which is thus proven true with the reflexive property.

Lessons from the Assignments

Lesson from Week 1

Week one was our review and introduction to the math side of what we'll be learning this semester. We started by revisiting the basic rules of discrete math. This meant getting back into the flow of writing out our proofs using the rules that we have access to. With only natural numbers to start, we started looking at successors again and eventually proving our way toward addition.

For me, this was a needed refresher because it's been a moment since I took discrete math, and I'm unfamiliar with writing my proofs as code, which is a learning curve for me. I'm a very pen-to-paper mathematician, so thinking about math at the same time I'm trying to recall syntax for code is a challenge for me. That said, we went through eight levels of proofs, and I was able to begin to get the hang of it.

I'm looking forward to bridging the gap between my math knowledge and how I involve it when I code. Sometimes I feel like I have the education to understand the concepts, but I struggle to apply them when I'm coding. The speed in which I type out code is not as quick as how I think about what I'd like to apply. This first assignment was a nice intro to opening my eyes as to what it might look like to get faster at that and also write technical reports in a coding environment as well. By shifting everything I do—the code, the math, the reporting—into an IDE, I know that I'll be able to get more comfortable working in that environment.

Conclusion

In week one, we reviewed our discrete math knowledge and began coding proofs.