CPSC-354 Report

Rohm Tandon
Chapman University

07/01/2024

Abstract

This report contains assignments throughout the fall 2024 semester and is intended for the purpose of documenting my work and showing my progress in CPSC 354 - Programming Languages, taught by Jonathan Weinberg.

Contents

1 Introduction

This report, prepared for CPSC 354 - Programming Languages at Chapman University, is a comprehensive account of my academic voyage over the semester. It includes a detailed compilation of my notes, homework solutions, and critical reflections on the coursework. This report serves as a bridge between the theoretical knowledge imparted in lectures and the practical skills essential for future pursuits in both graduate studies and the software industry.

2 Week by Week

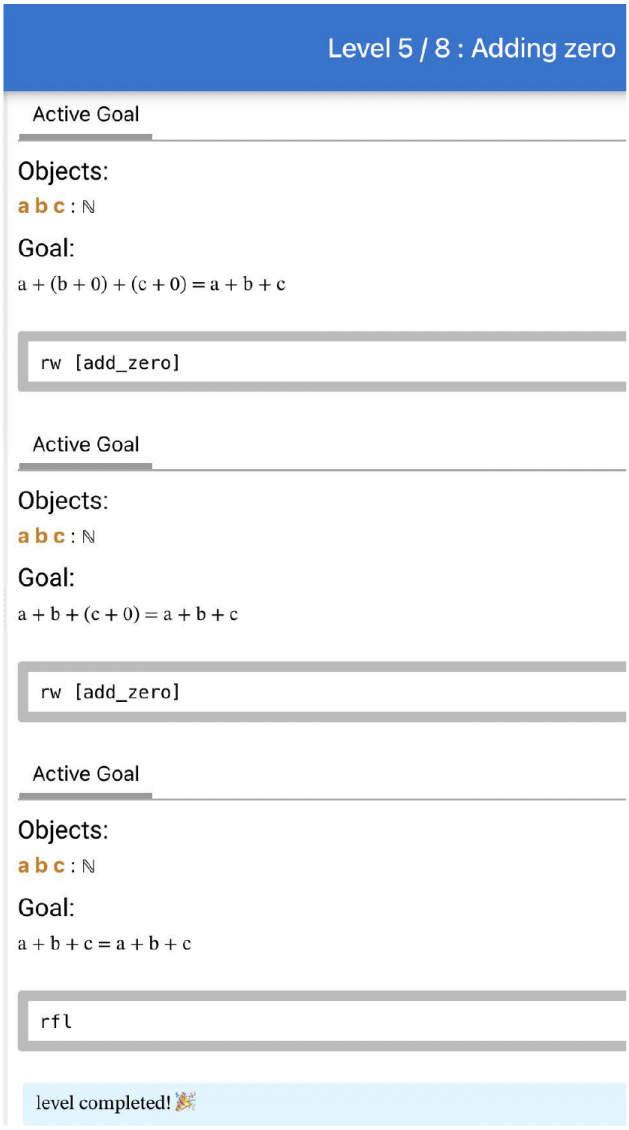
2.1 Week 1

Notes

In week 1 we learnt about Lean as a programming language and its correlation to discrete math. We also learnt about other proof assistants. We then shifted our focus to the NNG tutorial world as you can see below.

Homework

Tutorial world



Level 5:

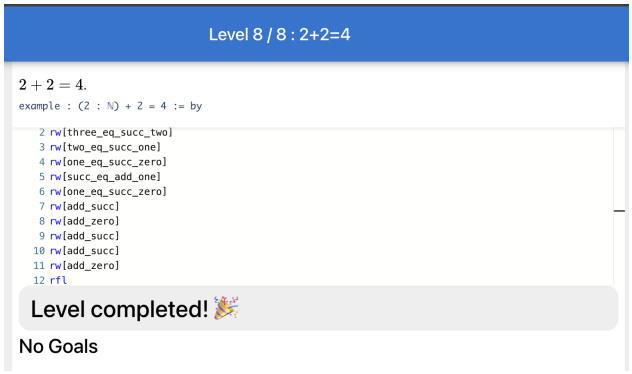
Discrete math's lemmas tell us that anything added to 0 will give the result of that number itself. So; A+0=A. Using this we can bring the left hand side down to a+b+c. From here we can use the property of reflexivity to show that both sides are equal, hence solving the puzzle.

Level 6:


Level 6 / 8 : Precision rewriting
Objects:
a b c : ℕ
Goal:
a + (b + 0) + (c + 0) = a + b + c
rw [add_zero c]
Active Goal
Objects:
a b c : ℕ
Goal:
a + (b + 0) + c = a + b + c
rw [add_zero b]
Active Goal
Objects:
a b c : ℕ
Goal:
a + b + c = a + b + c
rfl
level completed! 🎉

Level 7:

Level 8:



2.2 Week 2

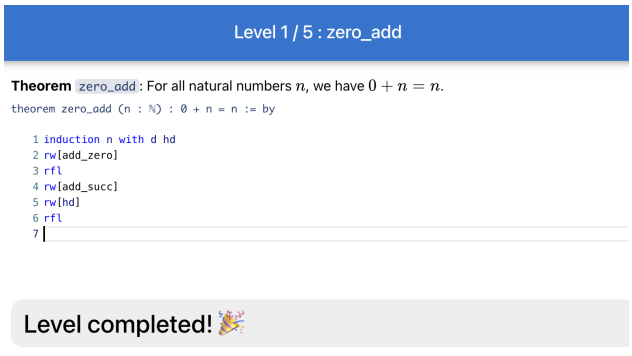
Notes

In week 2 we learnt about recursion and its application in other problems such as the Towers of Hanoi game we played. We also learnt about its various benefits such as breaking down complexity of problems and being more concise.

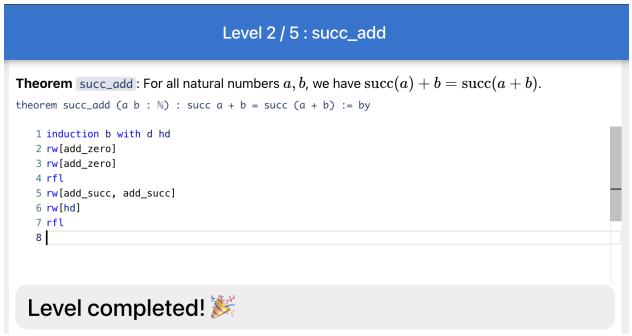
Homework

Addition world

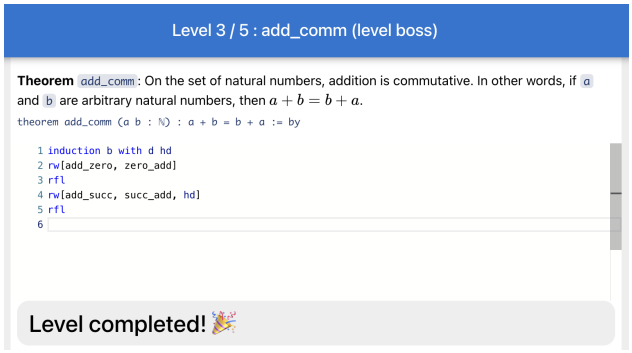
Level 1:



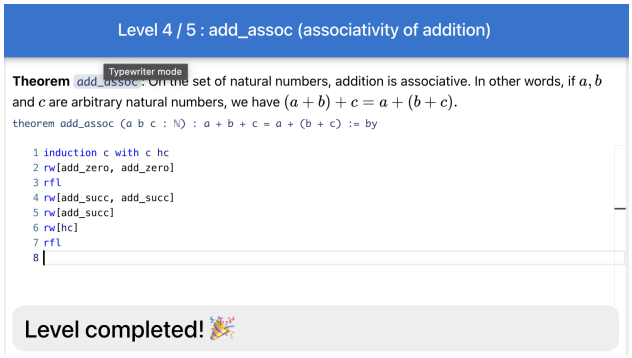
Level 2:



Level 3:

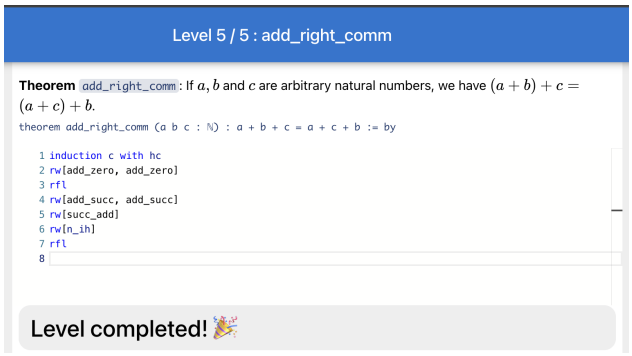


Level 4:



Using induction on c we can initially create an easier medium to use reflexivity to solve for a+b. Then solving the other side we just use the mathematical definiton of a successor function, until we can use the induction again to get the equation to the point where we can use reflexivity to prove it. This is a clear example of mathematical prrofs by induction.

Level 5:



Once again this is proof by mathematical induction similar to the previous one. This time we add the zeroes and then use reflexivity for the first part of the proof. Then to prove the second part we use the successor function until bringing back the induction we used like the previous question. Then to complete the proof we use reflexivity again.

-

Discord Question: Since recursion has so many benefits and also breaks down the complexity of problems, why aren't we taught to use it as our primary method? In other words, why isn't it the first method of problem solving we're taught?

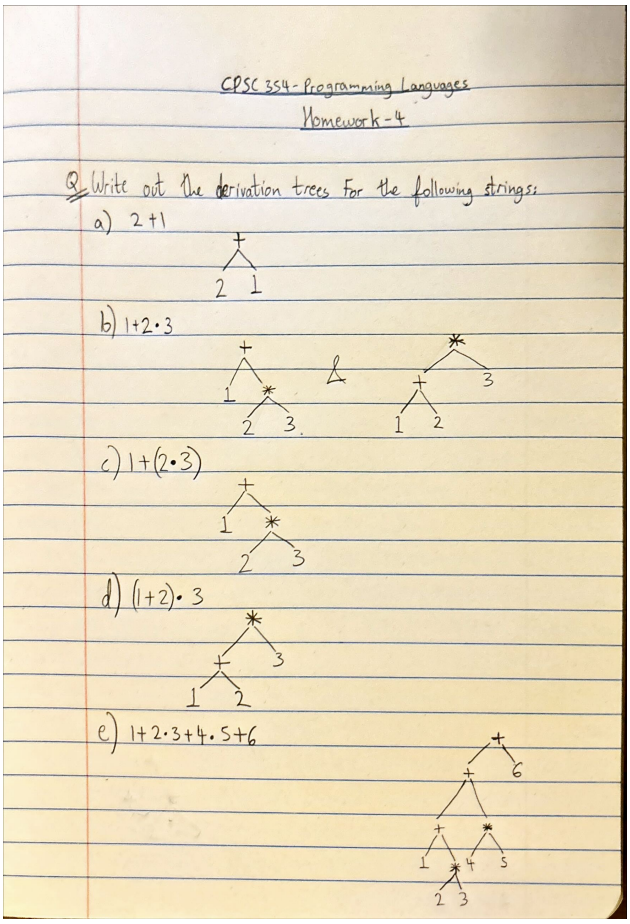
2.3 Week 3

Notes

In week 3 we spoke about recursion further, and focused on our calculators in python. Eventually connecting the dots for our first Assignment that was due at the same time as Homework 4, where I used recursion in my python calculator in order to get it to function as efficiently as possible. We also discussed parsing, and derivation trees which is what we practiced in Homework 4.

Homework

For homework 4 we did some practice on derivation trees for the strings you can see in the handwritten work below, along with the respective answers;



Discord question: Why do programming languages need different types of parsers, and how does this choice impact the way a computer understands the code?

3 Lessons from the Assignments

4 Conclusion

References

[BLA] Author, Title, Publisher, Year.