CPSC-354 Report

Your Name
Chapman University

September 8, 2024

Abstract

If will write this abstract... later!

Contents

1 Introduction

This introduction will be filled later, when I know what I want to say!

Grading guidelines (see also below):

- Is typesetting and layout professional?
- Is the technical content, in particular the homework, correct?
- Did the student find interesting references [BLA] and cites them throughout the report?
- Do the notes reflect understanding and critical thinking?
- Does the report contain material related to but going beyond what we do in class?
- Are the questions interesting?

Do not change the template (fontsize, width of margin, spacing of lines, etc) without asking your first.

2 Week by Week

2.1 Week 1

Notes

This week in class we mainly focused on learning Lean, setting up LaTeX, and a brief lecture on the basis of $Proof = Program$. This idea shows how logical mathematical proofs are a constructive process, building on previously founded theorems and definitions. This idea transfers to theoretical programming in the sense that programs are also constructed proofs. The execution of a program is the execution of many logical steps in a proof. I enjoy how this relates to the "human" process as well - our activities are the execution of previously learned strategies in a logical way.

Question

I was interested in learning more about Proof = Program, and some additional sleuthing showed me how a running a program is simply executing the steps in a logical proof. I was wondering how more complex program design methods such as recursion adhere to this idea, and if there are other programming methodologies that do not adhere to Proof = Program?

Homework

Level 5 - Adding Zero

In this level we prove the theorem that **a+0=a** using **a+(b+0)+(c+0)=a+b+c**. Here is how I solved this theorem.

repeat rw add zero

rfl

Level 6 - Adding Zero

In this level we built on the solution from the previous level to learn how to use precision rewriting.

rw add zero c

repeat rw add zero

rfl

Level 7 - Add Suc

In this level we prove the theorem that **succ(a)=a+1**.

rw one eq add zero

repeat rw add zero

rfl

Level 8 - Add Suc

In this level we prove the equation that **2+2=4**. This was the final level in the Tutorial World, and required the accumulation of definitions and theorems learned so far. I will provide the assumptions that I used when deciding my proof in Lean.

rw four eq succ three —— Any number $n = succ(pred(n))$

rw three eq succ two —— Any number $n = succ(pred(n))$

rw two eq succ one —— Any number $n = succ(pred(n))$

rw one eq succ zero —— Any number $n = succ(pred(n))$

repeat rw add succ —— Using $a + succb = succ(a + b)$

rw add zero —— Using $a + 0 = a$

rfl —— Proves the goal $X = X$

Comments and Questions

Ask at least one **interesting question**[1] on the lecture notes. Also post the question on the Discord channel so that everybody can see and discuss the questions.

2.2 Week 2

Notes

Translating Lean into Math by matching each line in Lean to the corresponding mathematical equation and assumptions. Being able to reverse the proof and translate from Math into Lean is also important. Lean reads from the goal to the axioms, whereas Math is written from the axioms to the goal (usually).

Defining data types recursively (in terms of itself). Syntax varies by language.

Recursion example with the Tower of Hanoi. Breaking logical puzzles into iterative steps, then into recursive steps.

Question

In class we used recursion to prove an algorithmic solution for the tower of hanoi game. Within the scope of proofs, what are the biggest drawbacks/advantages to using iterative vs recursive techniques?

Homework

Level 1 - Zero Add

In this level we prove the theorem that $\mathbf{0 + n = n}$. It was our first use of proof by induction.

induction n with d hd

rw add zero

rfl

rw add succ

rw hd

rfl

Level 2 - Succ Add

In this level we prove the theorem that $\mathbf{succ(a) + b = succ(a + b)}$.

induction b with d hd

repeat rw add zero

rfl

rw add succ

[1] It is important to learn to ask *interesting* questions. There is no precise way of defining what is meant by interesting. You can only learn this by doing. An interesting question comes typically in two parts. Part 1 (one or two sentences) sets the scene. Part 2 (one or two sentences) asks the question. A good question strikes the right balance between being specific and technical on the one hand and open ended on the other hand. A question that can be answered with yes/no is not an interesing question.

rw hd

rw add succ

rfl

Level 3 - Add Comm

In this level we prove the theorem that $\mathbf{a + b = b + a}$.

induction b

rw add zero

rw zero add

rfl

rw succ add

rw add succ

rw nih

rfl

Level 4 - Add Assoc

In this level we prove the theorem that $\mathbf{(a + b) + c = a + (b + c)}$.

induction b

rw add zero

rw zero add

rfl

rw add succ

rw succ add

rw nih

rw add comm

rw succ add

rw add succ

rfl

Level 5 - Add Comm Right

In this level we prove the theorem that $\mathbf{(a + b) + c = (a + c) + b}$. I will show the mathematical assumptions I used when deciding my proof in Lean.

rw add assoc a b —— (RHS) By associativity rule, $\mathbf{a + c + b = a + (c + b)}$

rw add assoc —— (LHS) By associativity rule, $\mathbf{a + b + c = a + (b + c)}$

rw add comm b c —— (LHS) By commutative rule, $\mathbf{a + (b + c) = a + (c + b)}$

rfl —— Proves the goal $\mathbf{X = X}$

3 Lessons from the Assignments

(Delete and Replace): Write three pages about your individual contributions to the project.

On 3 pages you describe lessons you learned from the project. Be as technical and detailed as possible. Particularly valuable are *interesting* examples where you connect concrete technical details with *interesting* general observations or where the theory discussed in the lectures helped with the design or implementation of the project.

Write this section during the semester. This is approximately a quarter of apage per week and the material should come from the work you do anyway. Just keep your eyes open for interesting lessons.

Make sure that you use LaTeX to structure your writing (eg by using subsections).

4 Conclusion

(Delete and Replace): (approx 400 words) A critical reflection on the content of the course. Step back from the technical details. How does the course fit into the wider world of software engineering? What did you find most interesting or useful? What improvements would you suggest?

References

[BLA] Author, Title, Publisher, Year.