

CPSC-354 Report

Your Name
Chapman University

September 15, 2024

Abstract

(Delete and Replace:) You can safely delete and replace the explanations in this file as they will remain available on the course website. For example, you should replace this abstract with your own. The abstract should be a short summary of the report. It should be written in a way that makes it possible to understand the purpose of the report without reading it.

Contents

1	Introduction	1
2	Week by Week	2
2.1	Week 1	2
2.2	Week 2	3
2.3	Week 3	6
3	Lessons from the Assignments	6
4	Conclusion	6

1 Introduction

(Delete and Replace): This report will document your learning throughout the course. It will be a collection of your notes, homework solutions, and critical reflections on the content of the course. Something in between a semester-long take home exam and your own lecture notes.¹

To modify this template you need to modify the source `report.tex` which is available in the course repo. For guidance on how to do this read both the source and the pdf of `latex-example.tex` which is also available in the repo. Also check out the usual resources (Google, Stackoverflow, LLM, etc). It was never as easy as now to learn a new programming lanugage (which, btw, \LaTeX is).

For writing \LaTeX with VSCode use the [LaTeX Workshop](#) extension.

There will be deadlines during the semester, graded mostly for completeness. That means that you will get the points if you submit in time and are on the right track, independently of whether the solutions are

¹One purpose of giving the report the form of lecture notes is that self-explanation is a technique proven to help with learning, see Chapter 6 of Craig Barton, *How I Wish I'd Taught Maths*, and references therein. In fact, the report can lead you from self-explanation (which is what you do for the weekly deadline) to explaining to others (which is what you do for the final submission). Another purpose is to help those of you who want to go on to graduate school to develop some basic writing skills. A report that you could proudly add to your application to graduate school (or a job application in industry) would give you full points.

technically correct. You will have the opportunity to revise your work for the final submission of the full report.

The full report is due at the end of the finals week. It will be graded according to the following guidelines.

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

In the reading about the mu puzzle, I learned about formal systems. The idea is that within the rules, you can reach the same desired outcome through different paths (at a given point you may have the option to apply several rules but can only chose one at any given step). In this context, a string which is created by applying one of the rules to the previous string is reffered to as a theorem.

I also learned about discrete mathematics proofs and look forward to learning about how they will be applied to the developement of a programing language.

Homework

NNG Tutorial World Level 5:

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

Solution:

First we use the Lean add zero proof to remove the 0 in b+0.

Then we use the Lean add zero proof to remove the 0 in c+0.

Finally we are left with $a+b+c = a+b+c$ and we can use rfl to confirm our proof with reflexive property.

...

NNG Tutorial World Level 6:

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

Solution:

First we use the Lean precision add zero proof (tarteting c) to remove the 0 in c+0.

Then we use the Lean add zero proof to remove the remaining 0 in b+0.

Finally we are left with $a+b+c = a+b+c$ and we can use rfl to confirm our proof with reflexive property.

...

NNG Tutorial World Level 7:

For all natural numbers a , we have $\text{succ}(a)=a+1$.

Solution:

First we unravel the one with the Lean rw proof to eliminate the one with a $\text{succ}0$.

Then we use the Lean rw proof with add succ to change $n + \text{succ}0$ into $\text{succ}(n+0)$.

Then we use the rw proof to rewrite $\text{succ}(n+0)$ into $\text{succ}(n)$.

Finally we are left with $\text{succ}(n) = \text{succ}(n)$ and we can use rfl to confirm our proof with reflexive property.

...

NNG Tutorial World Level 8:

For all natural numbers a , we have $\text{succ}(a)=a+1$.

Solution:

For this problem I simplified both sides of the equation using the rewrite proof with successors such as $3 = \text{succ } 2$ and more.

Eventually I got to this point: $\text{succ } (\text{succ } 0) + \text{succ } (\text{succ } 0) = \text{succ } (\text{succ } (\text{succ } (\text{succ } 0)))$

At this point I began using rw add succ to simplify the left side of the equation

Then I used rw add zero to remove a remaining zero and I was left with this: $\text{succ } (\text{succ } (\text{succ } (\text{succ } 0))) = \text{succ } (\text{succ } (\text{succ } (\text{succ } 0)))$

At this point I used the rfl to confirm my proof reflexively.

...

In all of the above examples I used the Lean rfl proof which directly corresponds to the mathematical reflexive property which states that: any number a is equal to itself. In other words $a = a$ or $b+c = b+c$, etc.

Comments and Questions

This section was a good refresh on some of the discrete mathematics concepts that I had forgotten over break. My question for this week is the following:

How are these mathematical concepts applied to the developement of programming languages?

2.2 Week 2

Notes

In the readings and class lectures about recursion and the towers of hanoi, I learned about what recursion is and how it can be used to optimize certain sysetems. This is evident in the towers of hanoi where we use a recursive formula or pattern to complete the challenge in the minimum amount of steps.

This was a good refresh on discrete mathematics and I look forward to learning how to apply the use of recursion in this course and in other challenges that can be solved mathematically.

Homework

NNG Addition World Level 1:

For all natural numbers n , we have $0 + n = n$.

Solution:

1. induction n with d hd
2. rw add zero
3. rfl
4. rw add succ
5. rw hd
6. rfl
- ...

NNG Addition World Level 2:

For all natural numbers a, b , we have $\text{succ}(a) + b = \text{succ}(a + b)$

Solution:

1. induction b
2. rw add zero
3. rfl
4. rw add succ
5. rw add succ
6. rw n ih
7. rfl
- ...

NNG Addition World Level 3:

On the set of natural numbers, addition is commutative. In other words, if a and b are arbitrary natural numbers, then $a + b = b + a$

Solution:

1. induction b
2. rw add zero
3. rw zero add
4. rfl
5. rw add succ
6. rw succ add
7. rw n ih
8. rfl
- ...

NNG Addition World Level 4:

On the set of natural numbers, addition is associative. In other words, if a, b and c are arbitrary natural numbers, we have $(a+b)+c=a+(b+c)$.

Solution:

1. induction a
2. rw zero add
3. rw zero add
4. rfl
5. rw succ add
6. rw succ add
7. rw succ add
8. rw n ih
9. rfl
- ...

NNG Addition World Level 5:

If a, b and c are arbitrary natural numbers, we have $(a+b)+c=(a+c)+b$.

Solution:

1. induction a
2. rw zero add
3. rw zero add
4. rw add comm
5. rfl
6. rw succ add
7. rw succ add
8. rw succ add
9. rw succ add
10. rw n ih
11. rfl

This lean proof is similar to the corresponding mathematical proof in the way that it uses induction to prove the theory on a single smaller equation and applies the inductive step and uses said proof to prove a property across all equations!

Comments and Questions

This section was yet another good refresh on discrete and reintroduced me to the concept of induction in proofs. I also loved the towers of hanoi activity and its modeling of recursive processes! My question is how similar puzzles can appear in programming languages and specifically the creation of them?

2.3 Week 3

Notes

This week we began developing our calculator in python which has proven to be more difficult than expected. I believe that the idea is to use a recursive method to scan an input for the subsections of a more complicated long input.

Homework

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 a page 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 L^AT_EX 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.