CPSC-354 Report

Max Hartel
Chapman University

September 1, 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

(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).

[1] One 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.

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 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

In week 1 we studied an introduction to discrete math, with a focus on natural numbers and understanding succession.

Notes and Homework

2.1.1 HW Problem 5:

Problem: We should define $a + 0$ to be a for any number a.

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

Assumption: $a, b, c \in \mathbb{N}$

Solution:

1. rw [add_zero b] $\rightarrow a + b + (c + 0) = a + b + c$
2. rw [add_zero c] $\rightarrow a + b + c = a + b + c$
3. rfl

Explanation:
On line 1, I apply the *add_zero* proof using the *rw* tactic to the variable b.
On line 2, I apply the *add_zero* proof using the *rw* tactic to the variable c.
On line 3, the equation is now identical on both sides, so I close the problem using reflexivity (*rfl*).

2.1.2 HW Problem 6:

Problem: Learn how to tell Lean to change $c + 0$ first by giving `add_zero` an explicit input.

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

Assumption: $a, b, c \in \mathbb{N}$

Solution:

1. rw [add_zero c]
2. rw [add_zero]
3. rfl

Explanation:
In this problem, we first tell Lean to simplify $c + 0$ by explicitly giving the *add_zero* proof to c. Then, we apply the *add_zero* proof to the remaining expression. Finally, since both sides of the equation are now identical, we conclude the proof using reflexivity (*rfl*).

2.1.3 HW Problem: 7

Problem: Let's prove that succ $n = n + 1$ for any natural number n.

Assumption: $n \in \mathbb{N}$

Solution:

1. rw [one_eq_succ_zero]	Apply the definition of 1: $1 = \text{succ } 0$
2. rw [add_succ]	Apply the add_succ lemma: $n + \text{succ } m = \text{succ } (n + m)$
3. rw [add_zero]	Simplify using add_zero: $n + 0 = n$
4. rfl	The equation now reflects, so the proof is complete.

Explanation:
In this proof, we proceed through the following steps:

1. We first rewrite 1 as succ 0 using the identity $1 = \text{succ } 0$.
2. Next, we apply the add_succ lemma, which states that $n + \text{succ } m = \text{succ } (n + m)$.
3. We then simplify using the add_zero lemma, which tells us that adding zero to a number n gives us n.
4. Finally, we conclude the proof with reflexivity (*rfl*), as both sides of the equation are identical.

Relation to Corresponding Mathematical Proof:
This proof exercise using lean relates the the mathematical proof of Natural numbers when we use the succession identity to state that 1 is equal to the succession of zero. This is only posible to do so because of the seconf postulate of natural numbers which states: "Predeccessors are Unique". Without the unique nature of predeccessors of natural numbers, it would not be possible to define a number by its natural successor like we do here. This postulate is stated on page 17 of 'Contemporary Discrete Mathematics' by Moshier.

2.1.4 HW Problem: 8

Problem: Let's prove that $2 + 2 = 4$ using Lean tactics.

Solution:

1. rw [two_eq_succ_one]	Rewrite 2 as succ 1
2. rw [add_succ]	Apply add_succ: $n + \text{succ } m = \text{succ } (n + m)$
3. rw [succ_eq_add_one]	Rewrite succ using $n + 1 = \text{succ } n$
4. rw [four_eq_succ_three]	Rewrite 4 as succ 3
5. rw [three_eq_succ_two]	Rewrite 3 as succ 2
6. rw [two_eq_succ_one]	Rewrite 2 as succ 1
7. repeat rw [succ_eq_add_one]	Apply the successor identity repeatedly to simplify the expression
8. rfl	The equation is now identical on both sides, completing the proof.

Explanation:
In this proof, we proceed step by step as follows:

1. We start by rewriting 2 as succ 1.
2. Then, we apply the `add_succ` lemma to handle the addition of a successor.
3. We rewrite the successor using the identity succ $n = n + 1$.
4. We then rewrite 4 as succ 3 and continue this process for 3 and 2.
5. The `succ_eq_add_one` lemma is applied repeatedly to reduce the expression to a simpler form.
6. Finally, we close the proof using reflexivity (*rfl*) because both sides of the equation are now equal.

Homework

(Delete and Replace:) This section will typically contain Homework problems. You should write up your solutions in LaTeX. You can use the `lstlisting` environment to include code. You can use Excalidraw for drawings. Pictures from handwritten drawings are acceptable if the drawings are of high quality (pictures from rough notes and quick sketches are likely to loose you points).

Make sure that this section can be read without referring back to the homework question. Introduce the question/problem and repeat it in your own words. Make sure to typset your homework in a way that makes it clear what the question and what the answer is. Present it as a worked example would be presented in a textbook.

Also explain what you learn from the homework. Each homework was carefully drafted to bring home a particular teaching point. Make sure to explain what this point is. Relate it to the big questions mentioned above.

Comments and Questions

Summary: This week we had an introduction to the course in general, and then we moved on to an introduction and review of discrete math. We focused on learning to prove basic theories about natural numbers and addition using an online system. This week we discussed how discrete math is very similar in programming in its structure. I especially thought it was interesting in the lecture notes when it differentiated between a definition of a process and an algorithm. Where I see definitions being utilized more in discrete math and algorithms being utilized more in computer science.

Question of the week: I have recently studied a boom in the Materials Science sector in the last year that came from AI led research methods, in which they were able to generate 10x more materials in a year than had ever before been created by having a deep learning network come up with potential combinations for new stable chemical formulas. Because of the speed and persistince of the network, it was able to vastly outperform human counterparts. Materials science shares some common factors with math because every chemical formula starts with the most basic atomic building blocks, just as mathematical proofs start by establinsing the most basic mathematical building blocks and then adding on from there. By applying the same methods emphasizing the speed and persistence of deep learning networks, can we expect a similar boom in new mathematical theories in the near future?

2.2 Week 2

(Delete:) Week 2 (and all the other weeks) should follow the same pattern as Week 1. Even if there is a week without homework, notes and comments (see above) are still expected.

2.3 ...

...

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.