CPSC-354 Report

Sarah Yoon
Chapman University

September 1, 2024

Abstract

Contents

1 Introduction

2 Week by Week

2.1 Week 1

Notes

Learned about some tactics and theorems

rfl: a tactic that proves theorems that take the form of X = X

rw: a tactic that rewrites a proof

one_eq_succ_zero: a theorem that proves 1 = succ 0 (there are also other similar existing theorems like two_eq_succ_one and so on)

add_zero: a theorem that proves a + 0 = a.

add_succ: a theorem that proves a + succ b = succ(a + b)

succ_eq_add_one: a theorem that proves succ a = a + 1

Homework

Problem 5:
a b c are in the set of natural numbers.
Prove that both sides are equal to each other.
a + (b + 0) + (c + 0) = a + b + c
rw [add_zero] - uses the add_zero theorem to prove that b + 0 = b
This is rewritten to:
a + b + (c + 0) = a + b + c
rw [add_zero] - this is done again to prove that c + 0 = c
This is rewritten to:
a + b + c = a + b + c
rfl - this proves that both sides that look the same are equal to each other

Problem 6:
This is the same problem as 5 but will be approached in a different manner.
a + (b + 0) + (c + 0) = a + b + c
rw [add_zero c] - specifically applies the add_zero theorem to c, making c + 0 into c
This is rewritten to:
a + (b + 0) + c = a + b + c
rw [add_zero b] - specifically applies the add_zero theorem to b, making b + 0 into b
This is rewritten to:
a + b + c = a + b + c
rfl - this proves that both sides that look the same are equal to each other

Problem 7:
n is in the set of natural numbers.
Prove that both sides are equal to each other.
succ n = n + 1 rw[one_eq_succ_zero] - rewrite 1 into successor 0 This is rewritten to:
succ n = n + succ 0
rw[add_succ] - uses the add_succ theorem to change n + succ 0 into succ(n + 0)
This is rewritten to:
succ n = succ (n + 0)
rw[add_zero] - uses the add_zero theorem to prove that n + 0 = n
This is rewritten to:
succ n = succ n
rfl - this proves that both sides that look the same are equal to each other

Problem 8:
Prove that both sides are equal to each other.
2 + 2 = 4
rw[two_eq_succ_one] - rewrites 2 into succ 1
This is rewritten to:
succ 1 + succ 1 = 4
rw[one_eq_succ_zero] - rewrites 1 into succ 0
This is rewritten to:
succ (succ 0) + succ (succ 0) = 4
rw[four_eq_succ_three] - rewrites 4 into succ 3
This is rewritten to:
succ (succ 0) + succ (succ 0) = succ 3

rw[three_eq_succ_two] - rewrites 3 into succ 2
This is rewritten to:
succ (succ 0) + succ (succ 0) = succ (succ 2)
rw[two_eq_succ_one] - rewrites 2 into succ 1
This is rewritten to:
succ (succ 0) + succ (succ 0) = succ (succ (succ 1))
rw[one_eq_succ_zero] - rewrites 1 into succ 0
This is rewritten to:
succ (succ 0) + succ (succ 0) = succ (succ (succ (succ 0)))
rw[add_succ] - changes succ (succ 0) + succ (succ 0) into succ (succ (succ 0) + succ 0)
This is rewritten to:
succ (succ (succ 0) + succ 0) = succ (succ (succ (succ 0)))
rw[add_succ] - changes succ (succ (succ 0) + succ 0) into succ (succ (succ (succ 0) + 0))
This is rewritten to:
succ (succ (succ (succ 0) + 0)) = succ (succ (succ (succ 0)))
rw[add_zero] - changes succ (succ 0) + 0 into
This is rewritten to:
succ (succ (succ (succ 0))) = succ (succ (succ (succ 0)))
rfl - this proves that both sides that look the same are equal to each other

For level 5: add_zero is a Lean proof that a + 0 = a (a representing any number). In mathematics, there are laws for arithemic. One of them is called the identity which applies to addition and multiplication. For addition, it states that m + 0 = m = 0 + m. This is the exact same as the Lean proof, a + 0 = a, which can also be written as a = 0 + a.

Comments and Questions

Learning the root of mathematics is very eye-opening, and I am confident it will be the same for programming languages. It provides another perspective for elementary functions like 2 + 2 equals 4, which is different from just knowing it through memorization. I feel as though this is why people have been able to expand mathematically. This makes me wonder: how can looking through the core of programming help us better current languages (e.g. python, rust)?

2.2 Week 2

2.3 ...

...

3 Lessons from the Assignments

4 Conclusion

References

[BLA] Author, Title, Publisher, Year.