Project 1 Report

Kyle Peppe

January 11, 2020

Abstract

This project was as an introduction to the ML programming language and some of the more basic commands that entails. I also then put our findings, code and output into Latex to further enhance our knowledge. I did the exercises from the PDF book from the class 2-5-1, 3-4-1, and 3-4-2. Below are the sections that are in this report for each problem:

- Problem statement
- Relevant code
- Test results

For each problem or exercise-oriented chapter in the main body of the report is a corresponding chapter in the Appendix containing the source code in ML. This source code is not pasted into the Appendix. Rather, it is input directly from the source code file itself.

Acknowledgments: I would like to acknowledge the 2 professors, Professor Chin and Professor Hamner, that have helped me begin to understand this new ML programming language. Also to Syracuse University for accepting me to this Masters program in Cybersecurity.

Contents

Chapter 1

Executive Summary

All requirements for this project are satisfied. Specifically,

Report Contents

The report has the following content:

Reproducibility in ML and LaTeX

Our ML and LaTeX source files compile with no errors.

Chapter 2

Exercise 2.5.1

2.1 Problem Statement for Exercise 2.5.1

For this problem I solved the Exercise 2.5.1 from the PDF book. This problem was to create a variable called timesPlus and then pass variables into the equation in order to prove the code was correct.

2.2 Relevant Code for Exercise 2.5.1

The relevant code wil be in the corresponding Appendix at the end of the report.

2.2.1 Test cases for Exercise 2.5.1

The relevant code wil be in the corresponding Relevant Information Chpater at the end of the report.

2.3 Test Results for Exercise 2.5.1

Below are the results from running the test cases:

The following is output from *ex-2-5-1.sml*

```
-----------------------------------------------------------------------
       HOL-4 [Kananaskis 11 (stdknl, built Sat Aug 19 09:30:06 2017)]

       For introductory HOL help, type: help "hol";
       To exit type <Control>-D
-----------------------------------------------------------------------
> > > > > val timesPlus = fn: int -> int -> int * int
> timesPlus 100 27;
timesPlus 10 26;
timesPlus 1 25;
timesPlus 2 24;
timesPlus 30 23;
timesPlus 50 200;
val it = (2700, 127): int * int
> val it = (260, 36): int * int
> val it = (25, 26): int * int
> val it = (48, 26): int * int
> val it = (690, 53): int * int
> val it = (10000, 250): int * int
>
```

Chapter 3

Exercise 3.4.1

3.1 Problem Statement for Exercise 3.4.1

For this problem I solved the Exercise 3.4.1 from the PDF book. This problem was to create a list and then using the cons operator gradually split up the lists into different lists. The intial list had 4 pairs with a index number and then a person's name. I did use a temp variable to split up the last values in order to continue to use the cons structure.

3.2 Relevant Code for Exercise 3.4.1

The relevant code wil be in the corresponding Appendix at the end of the report.

3.2.1 Test cases for Exercise 3.4.1

The relevant code wil be in the corresponding Relevant Information Chpater at the end of the report.

3.3 Test Results for Exercise 3.4.1

Below are the results from running the test cases:
The following is output from *ex-3-4-1.sml*

```
---------------------------------------------------------------------
        HOL-4 [Kananaskis 11 (stdknl, built Sat Aug 19 09:30:06 2017)]

        For introductory HOL help, type: help "hol";
        To exit type <Control>-D
---------------------------------------------------------------------
> > > > val x1 = 0: int
val x2 = 1: int
val x3 = 3: int
val x4 = 4: int
val y1 = "Alice": string
val y2 = "Bob": string
val y3 = "Carol": string
val y4 = "Dan": string
> val listA =  [(x1, y1), (x2, y2), (x3, y3), (x4, y4)];
val listA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:
   (int * string) list
> val (e1B :: listB) = listA;
val e1B = (0, "Alice"): int * string
val listB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
> val (e1C1, e1C2) = e1B;
val e1C1 = 0: int
val e1C2 = "Alice": string
> val (e1C3 :: temp) = listB;
val e1C3 = (1, "Bob"): int * string
val temp = [(3, "Carol"), (4, "Dan")]: (int * string) list
> val (e1C4 :: e1C5) = temp;
val e1C4 = (3, "Carol"): int * string
val e1C5 = [(4, "Dan")]: (int * string) list
>
```

Chapter 4

Exercise 3.4.2

4.1 Problem Statement for Exercise 3.4.2

For this problem I solved the Exercise 3.4.2 from the PDF book. This problem was to copy the code provided in the book and then to run it. Once we did that in order we had to explain the findings. I did this in the source code (surrounded by comments so the code should still compile).

4.2 Relevant Code for Exercise 3.4.2

The relevant code wil be in the corresponding Appendix at the end of the report.

4.2.1 Test cases for Exercise 3.4.2

The relevant code wil be in the corresponding Relevant Information Chpater at the end of the report.

4.3 Test Results for Exercise 3.4.2

Below are the results from running the test cases:
The following is output from *ex-3-4-2.sml*

```
---------------------------------------------------------------------
       HOL-4 [Kananaskis 11 (stdknl, built Sat Aug 19 09:30:06 2017)]

       For introductory HOL help, type: help "hol";
       To exit type <Control>-D
---------------------------------------------------------------------
> > > > val x1 = 1: int
val x2 = true: bool
val x3 = "Alice": string
> val pair1 = (x1, x3);
val pair1 = (1, "Alice"): int * string
> val list1 = [0, x1, 2];
val list1 = [0, 1, 2]: int list
> val list2 = [x2, x1];
poly: : error: Elements in a list have different types.
   Item 1: x2 : bool
   Item 2: x1 : int
   Reason:
      Can't unify bool (*In Basis*) with int (*In Basis*)
         (Different type constructors)
Found near [x2, x1]
Static Errors
> val list3 = (1 :: [x3]);
```

```
poly: : error: Type error in function application.
   Function: :: : int * int list -> int list
   Argument: (1, [x3]) : int * string list
   Reason:
      Can't unify int (*In Basis*) with string (*In Basis*)
         (Different type constructors)
Found near (1 :: [x3])
Static Errors
>
```

Chapter 5

Relevant Information

5.1 Specific Test Cases

5.1.1 Test Cases for Exercise 2-5-1

The required tests from the PDF project 1 handout are below:

```
(****************************************************************************)
(* Test Cases *)
(****************************************************************************)
timesPlus 100 27;
timesPlus 10 26;
timesPlus 1 25;
timesPlus 2 24;
timesPlus 30 23;
timesPlus 50 200;
```

5.1.2 Test Cases for Exercise 3-4-1

The required test cases as shown in the PDF project 1 handout (and parts a, b, and c in the textbook).

```
(a) The list of pairs assigned to listA
(b) The ML expression that results in the assignment of values to e1B
and listB as specified
(c) The ML expressions that result in the assignment of values to elC1,
elC2, elC3, elC4, and elC5 as specified
```

5.1.3 Test Cases for Exercise 3.4.2

For this exercise there was no actual test cases we just had to run the provided code and explain the output.

Appendix A

Source Code for Exercise 2.5.1

The following code is from *ex-2-5-1.sml*

```
(* Name:   Kyle Peppe                      *)
(* Email: kpeppe@syr.edu                   *)
(* ex-2-5-1.sml                            *)

(* Setting the timesPlus variable          *)
fun timesPlus x y = (x*y, x+y);

(***************************************)
(* Test Cases                              *)
(***************************************)
timesPlus 100 27;
timesPlus 10 26;
timesPlus 1 25;
timesPlus 2 24;
timesPlus 30 23;
timesPlus 50 200;
```

Appendix B

Source Code for Exercise 3.4.1

The following code is from *ex-3-4-1.sml*

```
(**************************************************************************)
(* Exercise 3.4.1                                                         *)
(* Author: Kyle Peppe                                                     *)
(* Date: 1/9/2020                                                         *)
(**************************************************************************)

(* Part A                                  *)
(* New List of Pairs                       *)
val [(x1, y1), (x2, y2), (x3, y3), (x4, y4)] = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4

(* Setting listA to values above           *)
val listA =  [(x1, y1), (x2, y2), (x3, y3), (x4, y4)];
(* End of Part A                           *)

(* Part B                                  *)
val (e1B :: listB) = listA;
(* End of Part B                           *)

(* Part C                                  *)
val (e1C1, e1C2) = e1B;
val (e1C3 :: temp) = listB;
val (e1C4 :: e1C5) = temp;
(* End of Part C                           *)
```

Appendix C

Source Code for Exercise 3.4.2

The following code is from *ex-3-4-2.sml*

```
(*************************************************************************)
(* Exercise 3.4.2                                                        *)
(* Author: Kyle Peppe                                                    *)
(* Date: 1/9/20                                                          *)
(*************************************************************************)

(* Code from the text book                                               *)
val (x1, x2, x3) = (1, true,"Alice");
val pair1 = (x1, x3);
val list1 = [0, x1, 2];
val list2 = [x2, x1];
val list3 = (1 :: [x3]);

(* Part 3 Answer                                                         *)
(* The first  error is happening because we are trying to create a list  *)
(* with a mixture of a boolean value and a numeric value. The 2nd error  *)
(* is similar except this time it is because of a numeric and string     *)
(* value trying to be combined into a list.                              *)
```