Project 2

Jinhao Wei

January 29, 2019

Abstract

This project is mainly a practice on programming in ML. In this project, we conducted several basic operations, such as pattern matching, in HOL. This report is basically a summary of each of the practice problems. For each of the problems, this report contains:

- Problem Statements
- Relevant Code
- Test Results (and Analysis)

In this report, we used several packages for formatting the document, including:

- a style file for the course, *634format.sty*
- the *listing* package for displaying and inputting ML source code
- *holtex.sty* and *holtexbasic.sty*, which was provided in professor Shiu-Kai Chin's sample report

Acknowledgments: This project followed Professor Shiu-Kai Chin's sample documents' format, and used multiple *.sty* files, such as *634format.sty*, *holtex.sty*, *holtexbasic.sty*, in Professor Shiu-Kai Chin's sample report.

Contents

Chapter 1

Executive Summary

All requirements for this project are satisfied Specifically,

Report Contents
This report consists of:

Reproducibility in ML and LaTeX
All ML and LaTeX source files compile well on the environment provided by this course.

Chapter 2

Exercise 2.5.1

2.1 Problem Statement

Our basic tasks in this problem are to:

1. Create a new *.sml* file, which contains code and test cases
2. Evaluate values using the function defined in code

2.1.1 Functions to implement

Our basic goal in this exercise is to implement and test a function named *timesPlus*, which takes in two integer values and returns a pair of integer values.

$$timesPlus\ x\ y = (x \times y, x + y)$$

2.1.2 Test cases

According to project requirements, we should run our functions on test cases below:

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

2.2 Relevant Code

We will define our function *timesPlus* as below:

```
(*  Name: Jinhao Wei      *)
(*  Email: jwei15@syr.edu *)
fun timesPlus x y = (x*y, x+y);
```

2.3 Test Results

If we send the above relevant code and test cases regions in the HOL, we will see a transcipt as below

1

```
---------------------------------------------------------------------
       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
> > # # # 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
> >
*** Emacs/HOL command completed ***

>
```

Chapter 3

Exercise 3.4.1

3.1 Problem Statement

In this problem, we focus on pattern matching in ML. We will use pattern matching to exact values from list and tuple. Our specific tasks are to produce expressions that

1. assign values to *listA*, a list of pairs
2. assign values to $e1B$, a pair, and *listB*, a list of pairs
3. assign values to $elC1$, $elC1$, $elC2$, $elC3$, $elC4$ and $elC5$ as they are specified

3.2 Relevant Code

```
(***********************************************************************)
(*          Exercise: 3.4.1                                            *)
(*          Author: Jinhao Wei                                         *)
(*          Date: January 29, 2019                                     *)
(***********************************************************************)
val listA = [(0,"Alice"), (1, "Bob"), (3, "Carol"), (4,"Dan")];
val elB::listB = listA;
val (elC1, elC2) = elB;
val (elC3::(elC4::(elC5::[]))) = listB;
```

3.3 Test Results

If we send the above region to HOL, we will see result as below:

```
> > > > val listA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:                1
   (int * string) list
val elB = (0, "Alice"): int * string
val listB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
val elC1 = 0: int
val elC2 = "Alice": string
val elC3 = (1, "Bob"): int * string
val elC4 = (3, "Carol"): int * string
val elC5 = (4, "Dan"): int * string
val it = (): unit
>
*** Emacs/HOL command completed ***
```

Chapter 4

Exercise 3.4.2

4.1 Problem Statement

In this problem, we will set serveral values in emacs, and evaluate them respectively under HOL environment. Finally we will check and analyze the results.

4.2 Relevant Code

```
(***************************************************************************)
(*   Exercise: 3.4.2                                                       *)
(*   Author: Jinhao Wei                                                    *)
(*   Date: January 29, 2019                                                *)
(***************************************************************************)

val (x1,x2,x3) = (1,true,"Alice");
val pair1 = (x1,x3);
val list1 = [0,x1,2];
val list2 = [x2,x1];
(*The line above should fail*)
(*x2 and x1 have different data type so they don't form a list*)

val list3 = (1 :: [x3]);
(*The line above should fail*)
(*int value can't be inserted to the front of a char list*)
```

4.3 Test Results and Analysis

According to the requirement, we should evaluate the values one by one and check the results. Therefore, I did not send the whole region to HOL, I ran each of them one after another.

The first three evaluation was quite successful. The first case is just a simple pattern matching. The second case is a tuple which can receive any different data types. As for the third case, all items in the list have the same data type, so the evaluation was successful.

```
> > > > val (x1,x2,x3) = (1,true,"Alice");                                    1
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
>
```

The following case failed to be evaluated, this is because all elements in a list should have the same data type. Howerver, from previous calculation we know that $x1$ is of type *int* while $x2$ is *bool*. So they can not make a list.

```
> val list2 = [x2,x1];                                                    2
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]
```

The following case also failed, this is because 1 is a data of type *int* while $[x3]$ is a list of *string*. Therefore the calculation failed.

```
>                                                                         3
> 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
```

Appendix A

Source Code for Exercise 2.5.1

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

```
(*   Name: Jinhao Wei        *)
(*   Email: jwei15@syr.edu  *)
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: Jinhao Wei                                         *)
(*          Date: January 29, 2019                                     *)
(***********************************************************************)
val listA = [(0,"Alice"), (1, "Bob"), (3, "Carol"), (4,"Dan")];
val elB::listB = listA;
val (elC1, elC2) = elB;
val (elC3::(elC4::(elC5::[]))) = listB;
```

Appendix C

Source Code for Exercise 3.4.2

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

```
(***********************************************************************)
(*  Exercise: 3.4.2                                                    *)
(*  Author: Jinhao Wei                                                 *)
(*  Date: January 29, 2019                                             *)
(***********************************************************************)

val (x1,x2,x3) = (1,true,"Alice");
val pair1 = (x1,x3);
val list1 = [0,x1,2];
val list2 = [x2,x1];
(*The line above should fail*)
(*x2 and x1 have different data type so they don't form a list*)

val list3 = (1 :: [x3]);
(*The line above should fail*)
(*int value can't be inserted to the front of a char list*)
```