

# Report for Simple ML Example

**Xiaozhi Li**

**13 September 2017**

## Abstract

The project brings together elements of functional programming in ML and documentation in  $\text{\LaTeX}$ . The purpose of this project is to lay the groundwork for credibility: results that are thoroughly documented and easily reproducible by independent third parties. We establish the documentation and programming infrastructure where each chapter documents a problem or exercise. Within each chapter are sections stating or showing:

- 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. This means changes in source code are easily captured in the report by recompiling the report in  $\text{\LaTeX}$ .

We introduce the use of style files and packages. Specifically, we use:

- a style file for the course, *634format.sty*,
- the *listings* package for displaying and inputting ML source code, and
- HOL style files and commands to display interactive ML/HOL sessions.

Finally, we show how to:

- easily generate a table of contents for the report, and
- refer to chapter and section labels in our report.

There are numerous  $\text{\LaTeX}$  tutorials on the web, for example, <https://www.latex-tutorial.com>, is very accessible for beginners.

---

**Acknowledgments:** We gratefully acknowledge the hard work, trust, and dedication of our past students in the Syracuse University Cyber Engineering Semester and the Air Force Research Laboratory's Advanced Course (ACE) in Engineering Cybersecurity Boot Camp. They bridged dreams and reality. In addition, this report is generated following the guidance of the course CIS 400-CSBD taught by Professor Shiu-kai Chin.

---

# Contents

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Executive Summary</b>                      | <b>3</b>  |
| <b>2</b> | <b>Exercise 2.5.1</b>                         | <b>4</b>  |
| 2.1      | Problem Statement . . . . .                   | 4         |
| 2.2      | Relevant Code . . . . .                       | 4         |
| 2.3      | Test Cases . . . . .                          | 4         |
| 2.4      | Test Results . . . . .                        | 4         |
| <b>3</b> | <b>Exercise 3.4.1</b>                         | <b>5</b>  |
| 3.1      | Problem Statement . . . . .                   | 5         |
| 3.2      | Relevant Code . . . . .                       | 5         |
| 3.3      | Test Result . . . . .                         | 5         |
| <b>4</b> | <b>Exercise 3.4.2</b>                         | <b>6</b>  |
| 4.1      | Problem Statement . . . . .                   | 6         |
| 4.2      | Relevant Code . . . . .                       | 6         |
| 4.3      | Test Cases . . . . .                          | 6         |
| 4.3.1    | Explain of error . . . . .                    | 7         |
| <b>5</b> | <b>Appendix A: Exercise 2.5.1 Source Code</b> | <b>8</b>  |
| <b>6</b> | <b>Appendix B: Exercise 3.4.1 Source Code</b> | <b>9</b>  |
| <b>7</b> | <b>Appendix C: Exercise 3.4.2 Source Code</b> | <b>10</b> |

## Chapter 1

---

# Executive Summary

---

**All requirements for this project are satisfied.** Specifically,

### Report Contents

Our report has the following content: Our report has the following content:

- Chapter 1: Executive Summary

- Chapter 2: Exercise 2.5.1

  - Section 2.1: Problem Statement

  - Section 2.2: Relevant code

  - Section 2.3: Test results

- Chapter 3

  - Section 3.1: Problem Statement

  - Section 4.2: Relevant code

  - Section 3.3: Test results

- Chapter 4

  - Section 4.1

  - Section 4.2

  - Section 4.3

  - Section 4.3.1

- Chapter 5:Appendix A: Exercise 2.5.1 Source Cod

- Chapter 6:Appendix B: Exercise 3.4.1 Source Cod

- Chapter 7:Appendix C: Exercise 3.4.2 Source Cod

### Reproducibility in ML and $\text{\LaTeX}$

Our ML and  $\text{\LaTeX}$  source files compile with no errors.

## Chapter 2

---

# Exercise 2.5.1

---

## 2.1 Problem Statement

In this exercise we are to define in ML the following functions:

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

## 2.2 Relevant Code

The following code takes advantage of function definition using *fun* in ML, and *currying*, i.e., defining functions with multiple arguments as a sequence of functions. This supports partial evaluation.

```
fun timesPlus x y= (x*y, x+y);
```

## 2.3 Test Cases

The required test cases for *timesPlus* are as follows.

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

## 2.4 Test Results

```
> > > val ListA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:
      (int * string) list
> val ListB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
val elB = (0, "Alice"): int * string
> 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
>
```

1

## Chapter 3

## Exercise 3.4.1

## 3.1 Problem Statement

In this exercise we are to define in ML the following functions:

```
val ListA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")];
val elB :: ListB = ListA;
val (elc1, elc2) = elB;
val [elc3, elc4, elc5] = ListB;
```

## 3.2 Relevant Code

The following code takes advantage of function definition using *fun* in ML, and *currying*, i.e., defining functions with multiple arguments as a sequence of functions. This supports partial evaluation.

```
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 Result

```
> > > val ListA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:
      (int * string) list
> val ListB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
val elB = (0, "Alice"): int * string
> 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
>
```

1

## Chapter 4

---

## Exercise 3.4.2

---

### 4.1 Problem Statement

In this exercise we are to define in ML the following functions:

```

val (x1,x2,x3) = (1, true, "Alice");
val pair1 = (x1, x3);
val list1 = [0, x1, 2];
val list2 = [x2, x1];
val list3 = (1 :: [x3]);

```

### 4.2 Relevant Code

The following code takes advantage of function definition using *fun* in ML.

```

val (x1,x2,x3) = (1,true," Alice" );
val pair1 = (x1,x3);
val list1 = [0,x1,2];
val list2 = [x2,x1];
val list3 = (1 :: [x3]);
val ListA = [(0, " Alice"), (1,"Bob"), (3, " Carol"),(4,"Dan" )];
val elB :: ListB= ListA;
val (elc1 ,elc2)=elB;
val [elc3 ,elc4 ,elc5]=ListB;

```

### 4.3 Test Cases

The following are the test results



```

> > > val x1 = 1: int
val x2 = true: bool
val x3 = "Alice": string
> val pair1 = (1, "Alice"): int * string
> val list1 = [0, 1, 2]: int list
> 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
> 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
> > > val ListA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:
  (int * string) list
> val ListB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
val elB = (0, "Alice"): int * string
> 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
>

```

### 4.3.1 Explain of error

The errors occurred in 3.4.2 are because of type matching. Errors in `val list2` and `val list3`, is that `list2` referenced `x2` from `(x1,x2,x3) = (1,true,"Alice")`. Where `x2` is the type `bool`, and HOL can't put `bool` and numbers in one list. `List3` contains `x3` which is a string type and Hol will not put string type and int in one list; therefore we got typeerrors.

## Chapter 5

---

# Appendix A: Exercise 2.5.1 Source Code

---

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

```
(* Name: Xiaozhi Li *)  
(* Email: xli137@syr.edu *)
```

```
fun timesPlus x y = (x*y, x+y);
```

## Chapter 6

---

## Appendix B: Exercise 3.4.1 Source Code

---

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

```
(*****)
(* Exercise 3.4.1 *)
(* Author: Xiaozhi Li *)
(* Date: 2017/9/13 *)
(*****)

val ListA = [(0, "Alice"), (1,"Bob"), (3, "Carol"),(4,"Dan")];

(* pattern matching: *)
val elB :: ListB = ListA;

val (elc1 , elc2)=elB;

val [ elc3 , elc4 , elc5]=ListB;
```

## Chapter 7

---

## Appendix C: Exercise 3.4.2 Source Code

---

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

```
(*****)
(* Exercise 3.4.2 *)
(* Author: Xiaozhi Li *)
(* Date: <2017/9/13> *)
(*****)

val (x1,x2,x3) = (1,true,"Alice");
val pair1 = (x1,x3);
val list1 = [0,x1,2];
val list2 = [x2,x1];
val list3 = (1 :: [x3]);
(* There are errors in val list2 and val list3, *)
(* the reason is that list2 referenced x2 from (x1,x2,x3) = (1,true,"Alice"); *)
(* where x2 is the type bool, and HOL can't put bool and numbers in one list *)
(* list3 contains x3 which is a string type and Hol will not put string type and int in one
```