# Project 1

**Alfred Murabito**

March 22, 2020

## Abstract

This report details results for the following exercise from *Certified Security by Design Using Higher Order Logic*: 2.5.1, 3.4.1, and 3.4.2. We go over function definitions in ML as well as ML types and type errors.

**Acknowledgments**: I received no assistance with this project.

# Contents

Chapter 1

# Executive Summary

All requirements for this project have been satisfied. A description and the results of each exercise are detailed in this project report. Each exercise is detailed with a problem statement, relevant code, and execution transcripts as listed in requirements.

Chapter 2

# Exercise 2.5.1

## 2.1 Problem Statement

We define a function in ML and evaluate it in HOL. The function returns a tuple of the sum and product of two numbers.

## 2.2 Relevant Code

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

## 2.3 Test Cases

Test cases below as listed in the project1 requirements.

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

Reults below:

```
---------------------------------------------------------------------
       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
>
```

Chapter 3

# Exercise 3.4.1

## 3.1 Problem Statement

We explore val definitons using pattern matching on tuples and lists in this example. The execution of our value declarations shown below.

## 3.2 Relevant Code

```
(* Part A *)
val listA = [(0,”Alice”),(1,”Bob”),(3,”Carol”),(4,”Dan”)]

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

(* Part C *)
val (e1C1,e1C2) = e1B;
val [e1C3,e1C4,e1C5] = listB;
```

## 3.3 Test Cases

Results below:

```
---------------------------------------------------------------------
       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 listA = [(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]:
   (int * string) list
> val e1B = (0, "Alice"): int * string
val listB = [(1, "Bob"), (3, "Carol"), (4, "Dan")]: (int * string) list
> val e1C1 = 0: int
val e1C2 = "Alice": string
> val e1C3 = (1, "Bob"): int * string
val e1C4 = (3, "Carol"): int * string
val e1C5 = (4, "Dan"): int * string
>
```

Chapter 4

# Exercise 3.4.2

## 4.1 Problem Statement

In this exercise, we execute ML code and observe a couple type errors that occur in the following value definitons.

## 4.2 Relevant Code

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

## 4.3 Test Cases

The last two definitons produce errors because (“x2” and “x1”) and (1 and “x3”) are different types and a list must have all elements be of the same type.
Results below:

```
---------------------------------------------------------------------
       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 = (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
>
```

Appendix A

# Exercise 2.5.1 Source Code

```
(******************************************************************************)
(* Author: Alfred Murabito                                                    *)
(* Date: March 21                                                             *)
(* email: acmurabi@syr.edu                                                    *)
(******************************************************************************)

(* Exercise 2.5.1 *)

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

# Exercise 3.4.1 Source Code

```
(******************************************************************************)
(* Author: Alfred Murabito                                                    *)
(* Date: March 21                                                             *)
(* email: acmurabi@syr.edu                                                    *)
(******************************************************************************)
(* Exercise 3_4_1 *)

(* Part A *)
val listA = [(0,"Alice"),(1,"Bob"),(3,"Carol"),(4,"Dan")]

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

(* Part C *)
val (e1C1,e1C2) = e1B;
val [e1C3,e1C4,e1C5] = listB;
```

## Appendix C

# Exercise 3.4.2 Source Code

```
(******************************************************************************)
(* Author: Alfred Murabito                                                    *)
(* Date: March 21                                                             *)
(* email: acmurabi@syr.edu                                                    *)
(******************************************************************************)
(* Exercise 3.4.2 *)

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