# Project 1

Michael Hrishenko, LaTeX Journeyman

January 18, 2020

**Abstract**: Project one, purpose is mainly familiarization with HOL and Emacs, and also Latex reports. Multiple exercises and HOL and discussions concerning HOL errors, type constraints, and data structures & mapping.

**Acknowledgements**: Professors Shiu-Kai Chin & Susan Older

# Contents

# Chapter 1

# Executive Summary

All objectives complete. No problems to report other than lack of time due to operations tempo with Iranian tensions. Things have calmed and we have a weekend again, which I will use to catch up on work. Of note I am currently using Emacs run through the XServer on my Windows, while Emacs runs on Windows Subsystem for Linux (WSL) to avoid virtualization headaches. The setup is like a dream compared to the VM, which I realize is the catch-all solution but for those of us running Microsoft I believe this solution might be preferred. First time using Emacs so still some familiarization on that front. Otherwise straightforward and fully complete.

# Chapter 2

# Exercise 2.5.1

## 2.1 Problem Statement

1. Start up Emacs with a fresh file ex-2-5-1.sml.
2. Start HOL inside of Emacs, highlight the definition of timesPlus, and send the region to HOL.
3. Evaluate the expression timesPlus 100 27 within HOL. If you've done things correctly, you should get a pair of integers as a result. Note: when you start HOL within Emacs, a second window opens below or on the right of your source code. This is the *HOL* buffer. Move your cursor to this buffer by using your mouse or by typing C-x o, which moves the cursor among the various Emacs buffers/windows.
4. Kill the HOL process while preserving the *HOL* window by moving your cursor to the *HOL* window and typing C-D. Save the contents of the *HOL* window under the name ex-2-5-1.trans.

## 2.2 Code

```
(* Name: Michael Hrishenko *)
(* Email: mahrishe@syr.edu *)
fun timesPlus x y = (x*y, x+y);
```

## 2.3 Test Cases

1. Input:

   ```
   timesPlus 100 27;
   ```

2. Output:

   ```
   val it = (2700, 127): int * int
   ```

# Chapter 3

# Exercise 3.4.1

## 3.1 Problem Statement

Create a file ex-3-4-1.sml as your sourcefile. Define the following values in ML. Please include comments similar to those in the examples we have shown in this Chapter. Execute your final source code in the HOL interpreter and create a transcript file ex-3-4-1.trans by saving the *HOL* window in Emacs to ex-3-4-1.trans.

1. Devise the list of pairs [(0,"Alice"), (1,"Bob"), (3,"Carol"),(4,"Dan")] and assign it the name listA.

2. Using listA and pattern matching, create the following value assignments: elB has the value (0,"Alice") and listB has the value [(1,"Bob"),(3,"Carol"),(4,"Dan")]

3. Using elB, listB, and pattern matching, create the following value assignments: elC1 has the value 0, elC2 has the value "Alice", elC3 has the value (1, "Bob"), elC4 has the value (3, "Carol"), and elC5 has the value (4, "Dan").

## 3.2 Code

```
(*******************************************************)
(* Exercise 3.4.1 *)
(* Author: Michael Hrishenko *)
(* Date: 17JAN2020 *)
(*******************************************************)
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 Cases

1. Input:

   ```
   val (elB :: listB) = listA;
   ```

2. Output:

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

3. Input:

```
val (elC1, elC2) = elB;
```

4. Output:

```
val elC1 = 0: int
val elC2 = "Alice": string
```

5. Input:

```
val (elC3 :: elC4 :: elC5 :: []) = listB;
```

6. Output:

```
val elC3 = (1, "Bob"): int * string
val elC4 = (3, "Carol"): int * string
val elC5 = (4, "Dan"): int * string
```

# Chapter 4

# Exercise 3.4.2

## 4.1 Problem Statement

Create a file ex-3-4-2.sml as your sourcefile. Define the following values in ML. Please include comments similar to those in the examples we have shown in this Chapter. Execute your final source code in the HOL interpreter and create a transcript file ex-3-4-2.trans by saving the *HOL* window in Emacs to ex-3-4-2.trans.

1. Evaluate each of the assignments in the order in which they appear in HOL. Store the results in your ex-3-4-2.trans file.

2. Explain in your own words what the errors are that HOL detects. Include your answers as comments in your source code.

## 4.2 Code

```
(*******************************************************)
(* Exercise 3.4.2 *)
(* Author: Michael Hrishenko *)
(* Date: 17JAN2020 *)
(*******************************************************)
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

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

   - NO ERRORS

2. val pair1 = (x1,x3);

   - NO ERRORS

3. val list1 = [0,x1,2];

   - NO ERRORS

4. val list2 = [x2,x1];

   - Elements in a list have different types: Since x2 is "true" and x1 is '1', the list cannot be constructed. This is a HOL feature, whereas Python lists and other languages may hold elements

of different types, HOL is designed to root out type errors and prove assurance in design, so mixed types are not allowed.

5. val list3 = (1 :: [x3]);

   - NO ERRORS

# Chapter 5

# Appendices

## .1 Appendix A

```
(* Name: Michael Hrishenko *)
(* Email: mahrishe@syr.edu *)

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

## .2 Appendix B

```
(*******************************************************)
(* Exercise 3.4.1 *)
(* Author: Michael Hrishenko *)
(* Date: 17JAN2020 *)
(*******************************************************)

val listA = [(0,"Alice"), (1,"Bob"), (3,"Carol"),(4,"Dan")];
val (elB :: listB) = listA;
val (elC1, elC2) = elB;
val (elC3 :: elC4 :: elC5 :: []) = listB;
```

## .3 Appendix C

```
(*******************************************************)
(* Exercise 3.4.2 *)
(* Author: Michael Hrishenko *)
(* Date: 17JAN2020 *)
(*******************************************************)

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