

Report 1

Chirag Sachdev

Week 2

Abstract

This project is a part of HW2 of Assurance Foundations. The homework deals with integration of ML and HOL to L^AT_EX. The goal of this report is to show reproducibility which is the groundwork for credibility that I have done this on my own without any external help. Every Chapter demonstrates the following sections:

- Problem Statement
- Relevant Code
- Test Results

This project includes the following packages:

634format.sty A format style for this course

listings Package for displaying and inputting ML source code

holtex HOL style files and commands to display in the report

This document also demonstrates my ability to :

- Easily generate a table of contents,
- Refer to chapter and section labels

My skills and my professional details can be found at <https://www.linkedin.in/in/chiragsachdev>.

acknowledgments

I would gratefully acknowledge Dr. Shiu-Kai Chin and my other professors at Syracuse University and my Professors at Drexel University for being the wonderful mentors they are to guide me through my journey of obtaining a Master's Degree.

Contents

1	Executive Summary	3
2	Excercise 2.5.1	4
2.1	Problem statement	4
2.2	Relevant Code	4
2.3	Test Case	4
3	Excercise 3.4.1	6
3.1	Problem statement	6
3.2	Relevant Code	6
3.3	Test Case	7
4	Excercise 3.4.2	8
4.1	Problem statement	8
4.2	Relevant Code	8
4.3	Test Case	9
A	Source code for Ex 2.5.1	10
B	Source code for Ex 3.4.1	11
C	Source code for Ex 3.4.2	12

Chapter 1

Executive Summary

All requirements for this report have been met Specifically,

Report Contents The report has the following content:

- Chapter 1 Executive Summary

- Chapter 2 Exercise 2.5.1

- (a) Section 2.1: Problem Statement

- (b) Section 2.2 Relevant code

- (c) Section 2.3 Test Results

- Chapter 3 Exercise 3.4.1

- Section 3.1: Problem Statement

- Section 3.2: Relevant code

- Section 3.3: Test Results

- Chapter 3 Exercise 3.4.1

- Section 4.1: Problem Statement

- Section 4.2: Relevant code

- Section 4.3: Test Results

- Appendix A: Source Code:2.5.1

- Appendix B: Source Code:3.4.1

- Appendix C: Source Code:3.4.2

Chapter 2

Excercise 2.5.1

2.1 Problem statement

1. Start up Emacs with a fresh file *ex-2-5-1.sml*.
2. In Emacs, insert the following text into *ex-2-5-1.sml*, where *(** and **)* are used to surround comments in ML.

```

(* Name: fill in your name *)
(* Email: fill in your email address *)

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

3. Start HOL inside of Emacs,highlight the definition of *timesPlus*, and send the region to HOL.
4. 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, in a second window opens below or the right of your sourcecode. This the **HOL** buffer. Move your cursor to this buffer by using the mouse or by typing **C-x o**, which moves the cursor among the various Emacs buffer/windows.
5. Kill the HOL process while preserving the **HOL** window by moving the cursor to the **HOL** wondow and typing **C-d**. Save the contents of the **HOL** window under the name *ex-2-5-1.tans*.

2.2 Relevant Code

```
fun timesPlus x y = (x*y,x+y);
timesPlus 100 27
```

2.3 Test Case

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

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
>
Process HOL finished
```

Chapter 3

Excercise 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*. At the top of your *ex-3-4-1.sml* file, include the following comment block:

```
(*****
(* Exercise 4.4.1
(* Author: <your name>
(* Date: <date you wrote the file>
(*****)
```

Devise ML expressions for the following values and assign them to the constant names as specified.

- a Devise the list of pairs $[(0, "Alice"), (1, "Bob"), (3, "Carol"), (4, "Dan")]$ and as-sign it the name *listA*.
- b Using *listA* and pattern matching, create the following value assignments: *elB* has the value $(0, "Alice")$ and *listB* has the value $[(1, "Bob"), (3, "Carol"), (4, "Dan")]$
- c 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 Relevant Code

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

3.3 Test Case

2

```
-----
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 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 listB = [(3, "Carol"), (4, "Dan")]: (int * string) list
> val elC4 = (3, "Carol"): int * string
val listB = [(4, "Dan")]: (int * string) list
> val elC5 = (4, "Dan"): int * string
val listB = []: (int * string) list
> >
Process HOL finished
```

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*. At the top of your *ex-3-4-2.sml* file, include the following comment block:

```
(*****  
(* Exercise 4.4.2  
(* Author: <your name>  
(* Date: <date you wrote the file>  
(***)
```

1. Insert the following code into your *ex-3-4-2.sml* file:

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

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

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 Case

3

```
-----
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
>
Process HOL finished
```

Appendix A

Source code for Ex 2.5.1

```
(*Name: Chirag Sachdev*)
(*Email: cjsachde@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 Ex 3.4.1

```
(* ----- *)
(* Excercise 4.4.1 *)
(* Author: Chirag Sachdev *)
(* Date: 01/29/19 *)
(* ----- *)

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

Appendix C

Source code for Ex 3.4.2

```

(*****)
(* Exercise 4.4.2 *)
(* Author: Chirag Sachdev *)
(* Date: 01/29/2019 *)
(*****)

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

(* ----- *)
(* HOL lists have to be values of the same data type, *)
(* list2 cannot have a boolean and an integer type *)
(* Hence HOL gives an error saying it cannot unify the *)
(* constructors with different data types *)
(* ----- *)

val list2 = [x2,x1];

(* ----- *)
(* HOL lists have to be values of the same data type, *)
(* list3 cannot have an integer and a string type *)
(* Hence HOL gives an error saying it cannot unify the *)
(* constructors with different data types *)
(* x3 is of String data type and 1 is an integer *)
(* ----- *)

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

```