Report for Project5

Xiaozhi Li

October 6, 2017

Abstract

Project 5 using LaTeX. In this project we will use EmitTeX to generate the HOL reports. All HOL source files are included in the HOL folder.

Acknowledgments: This report follows the hand book Certified Security by Design Using Higer Order Logic, and course instructions from CIS400-CSBD.

Contents

Chapter 1

Executive Summary

All requirements for this project are satisfied. Specifically, we utilized HOL to prove the following theorems:

[conjSymAll]

$\vdash \forall p\ q.\ p \land q \iff q \land p$

[conjSymThm]

$\vdash p \land q \iff q \land p$

[problem1Thm]

$\vdash p \Rightarrow (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \Rightarrow r$

All requirments of the project have been met, all theories and code compiled and ran within HOL and Latex.

Chapter 2

Exercise 8.4.1

2.1 Problem Statement

Our task is to prove the theorem $\vdash p \Rightarrow (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \Rightarrow r$

2.2 HOL Code

```
(* 8-4-1  *)
val problem1Thm =
let
val th1 =ASSUME ``p:bool``
val th2 =ASSUME ``p ==> q``
val th3 =ASSUME ``q ==> r``
val th4 =MP th2 th1
val th5 =MP th3 th4
val th6 = DISCH (hd(hyp th3)) th5
val th7 = DISCH (hd(hyp th2)) th6

in
DISCH (hd(hyp th1)) th7
end
```

2.3 Session Transcript

```
> > > > # # # # # # # # # ** Unicode trace now off                              1
> *** Globals.show_assums now true ***
> # # # # # # # # # ** types trace now on
> # # # # # # # # # # # # val problem1Thm =
    [] |- (p :bool) ==> (p ==> (q :bool)) ==> (q ==> (r :bool)) ==> r:
   thm
>
```

2.4 Explain Result

Hol is showing our theorem with no type errors, this means our tests have passed.

Chapter 3

Exercise 8.4.2

3.1 Problem Statement

For 8.4.2 we need to prove the therom:

$\vdash p \land q \iff q \land p$

3.2 HOL Code

```
val conj1Thm =
let
val th2 =ASSUME ``p/\q``
val th3 =CONJUNCT1 th2
val th4 =CONJUNCT2 th2
val th5 =CONJ th4 th3
in
DISCH (hd(hyp th2)) th5
end;

val conj2Thm =
let
val th1 =ASSUME ``q/\p``
val th2 =CONJUNCT1 th1
val th3 =CONJUNCT2 th1

val th4 =CONJ th3 th2
in
DISCH (hd(hyp th1)) th4
end;


val conjSymThm =
IMP_ANTISYM_RULE conj1Thm conj2Thm;
```

3.3 Session Transcript

1

```
> > > > # # # # # # # # # # ** types trace now on
> *** Globals.show_assums now true ***
> # # # # # # # # # ** Unicode trace now off
> # # # # # # # # val conj1Thm =
    [] |- (p :bool) /\ (q :bool) ==> q /\ p:
   thm
> > # # # # # # # # # # val conj2Thm =
    [] |- (q :bool) /\ (p :bool) ==> p /\ q:
   thm
> > > # val conjSymThm =
    [] |- (p :bool) /\ (q :bool) <=> q /\ p:
   thm
>
*** Emacs/HOL command completed ***
```

3.4 Explain Result

In 8.4.2, all of our theorem and theory have passed by HOL.

Chapter 4

Excersice 8.4.3

4.1 Problem Statement

For 8.4.3 we need to prove the therom:

$\vdash \forall p\ q.\ p \wedge q \iff q \wedge p$

4.2 HOL Code

Notice in 8.4.3 we are extending the code from 8.4.2:

```
val conj1Thm =
let
val th2 =ASSUME ``p/\q``
val th3 =CONJUNCT1 th2
val th4 =CONJUNCT2 th2
val th5 =CONJ th4 th3
in
DISCH (hd(hyp th2)) th5
end;

val conj2Thm =
let
val th1 =ASSUME ``q/\p``
val th2 =CONJUNCT1 th1
val th3 =CONJUNCT2 th1

val th4 =CONJ th3 th2
in
DISCH (hd(hyp th1)) th4
end;


val conjSymThm =
IMP_ANTISYM_RULE conj1Thm conj2Thm;
val conjSymAll=GENL [``p:bool``, ``q:bool``] conjSymThm;
```

4.3 Session Transcript

1

```
> > > > # # # # # # # # # # ** types trace now on
> *** Globals.show_assums now true ***
> # # # # # # # # # ** Unicode trace now off
> # # # # # # # # val conj1Thm =
    [] |- (p :bool) /\ (q :bool) ==> q /\ p:
   thm
> > # # # # # # # # # val conj2Thm =
    [] |- (q :bool) /\ (p :bool) ==> p /\ q:
   thm
>
*** Emacs/HOL command completed ***

> # val conjSymThm =
    [] |- (p :bool) /\ (q :bool) <=> q /\ p:
   thm
> val conjSymAll =
    [] |- !(p :bool) (q :bool). p /\ q <=> q /\ p:
   thm
>
```

4.4 Explain Result

All tests from 8.4.3 have been passed in HOL.

Chapter 5

Appendix A: source code for 8.4.1, 8.4.2, and 8.4.3

The following code is from *proj5Script.sml*

```
(***************************************************************************)
(* Author: Xiaozhi Li                                                     *)
(***************************************************************************)

(***************************************************************************)
(* All HOL script files are ML modules, so we need to declare the file    *)
(* example1Script as an ML structure.  Do this with the "structure: command *)
(* as the very first executable line.  The very last executable line is "end" *)
(***************************************************************************)

structure proj5Script = struct

(***************************************************************************)
(* Note: everything after new_theory must be part of a val assignment, when *)
(* using Holmake.  Otherwise, there will be compilation errors. If you don't *)
(* want to assign an expression to a name, just use "val _ =<expression>"  *)
(* The "_" indicates that we don't want to have a name.                    *)
(***************************************************************************)
open HolKernel Parse boolLib bossLib;

val _ = new_theory "proj5";
(***************************************************************************)
(* val problem1Thm                                                         *)
(* [] |-p==> (p==>q) ==> (q==>r) ==>r                                      *)
(***************************************************************************)

(* 8-4-1  *)
val problem1Thm =
let
val th1 =ASSUME ``p:bool``
val th2 =ASSUME ``p ==> q``
val th3 =ASSUME ``q ==> r``
val th4 =MP th2 th1
val th5 =MP th3 th4
val th6 = DISCH (hd(hyp th3)) th5
val th7 = DISCH (hd(hyp th2)) th6

in
DISCH (hd(hyp th1)) th7
end
```

```
val _ =save_thm("problem1Thm",problem1Thm);







(* 8-4-2*)

val conj1Thm =
let
val th2 =ASSUME ``p/\q``
val th3 =CONJUNCT1 th2
val th4 =CONJUNCT2 th2
val th5 =CONJ th4 th3
in
DISCH (hd(hyp th2)) th5
end;

val conj2Thm =
let
val th1 =ASSUME ``q/\p``
val th2 =CONJUNCT1 th1
val th3 =CONJUNCT2 th1

val th4 =CONJ th3 th2
in
DISCH (hd(hyp th1)) th4
end;


val conjSymThm =
IMP_ANTISYM_RULE conj1Thm conj2Thm;

val _ =save_thm("conjSymThm",conjSymThm);


(*********)
(* 8-4-3 *)




val conj1Thm =
let
val th2 =ASSUME ``p/\q``
val th3 =CONJUNCT1 th2
val th4 =CONJUNCT2 th2
val th5 =CONJ th4 th3
in
```

```
DISCH (hd(hyp th2)) th5
end;

val conj2Thm =
let
val th1 =ASSUME ``q/\p``
val th2 =CONJUNCT1 th1
val th3 =CONJUNCT2 th1

val th4 =CONJ th3 th2
in
DISCH (hd(hyp th1)) th4
end;


val conjSymThm =
IMP_ANTISYM_RULE conj1Thm conj2Thm;
val conjSymAll=GENL [``p:bool``, ``q:bool``] conjSymThm;

val _=save_thm("conjSymAll", conjSymAll)

val _=export_theory();

end (* structure *)
```