Report for Project5

Xiaozhi Li

October 13, 2017

Abstract

Project 6 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

Not all requirements for this project are satisfied. Specifically, we utilized HOL to prove the following theorems:

[absorptionRule]

$\vdash \forall p\ q.\ (p \Rightarrow q) \Rightarrow p \Rightarrow p \wedge q$

[absorptionRule2]

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

[constructiveDilemmaRule]

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

[constructiveDilemmaRule2]

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

[problemOnethm]

$\vdash M\ s$

[problemTwothm]

$\vdash p \Rightarrow \neg q$

Exercise 10.4.3 was not included in this project.

Chapter 2

Exercise 9.5.1

2.1 Problem Statement

Our task is to prove the theorem

$\vdash \forall p\ q.\ (p \Rightarrow q) \Rightarrow p \Rightarrow p \land q$

2.2 HOL Code

```
(* 9-5-1 *)
val absorptionRule=

TAC_PROOF (
([],‘‘!p q. (p ==> q) ==> p ==> p/\q‘‘),
(REPEAT STRIP_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC) );

val _=save_thm(”absorptionRule”,absorptionRule);
val _=export_theory();
```

2.3 Session Transcript

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

2.4 Explain Result

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

Chapter 3

Exercise 9.5.2

3.1 Problem Statement

For 9.5.2 we need to prove the therom:

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

3.2 HOL Code

```
val constructiveDilemmaRule=

 TAC_PROOF (
([], ``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
REPEAT STRIP_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC []
);
```

3.3 Session Transcript

```
> # # # # # # # # # # val constructiveDilemmaRule =                                        1
    []
|- !(p :bool) (q :bool) (r :bool) (s :bool).
     (p ==> q) /\ (r ==> s) ==> p \/ r ==> q \/ s:
   thm
```

3.4 Explain Result

In 9.5.2, all of our theorem and theory have passed by HOL.

Chapter 4

Excersice 9.5.3

4.1 Problem Statement

For 9.5.3 we need to prove the therom:

$\vdash \forall p\ q.\ (p \Rightarrow q) \Rightarrow p \Rightarrow p \wedge q$

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

using PROVE_TAC .

4.2 HOL Code

In 9.5.3, our relative HOL code is:

```
(* 9-5-3 *)

val absorptionRule2=
 TAC_PROOF (
([], ``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
PROVE_TAC []
);

val _=save_thm("absorptionRule2",absorptionRule2);


val constructiveDilemmaRule2=
 TAC_PROOF (
([],``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
PROVE_TAC []
);
```

4.3 Session Transcript

```
> # # # # Meson search level: ................
val absorptionRule2 =
    []
|- !(p :bool) (q :bool) (r :bool) (s :bool).
     (p ==> q) /\ (r ==> s) ==> p \/ r ==> q \/ s:
   thm
> > > > > # # # # Meson search level: ................
val constructiveDilemmaRule2 =
    []
|- !(p :bool) (q :bool) (r :bool) (s :bool).
     (p ==> q) /\ (r ==> s) ==> p \/ r ==> q \/ s:
   thm
>
```

4.4 Explain Result

All tests from 9.5.3 have been passed in HOL.

Chapter 5

Exercise 9.5.2

5.1 Problem Statement

For 9.5.2 we need to prove the therom:

$\vdash \forall p\ q\ r\ s.\ (p \Rightarrow q) \wedge (r \Rightarrow s) \Rightarrow p \vee r \Rightarrow q \vee s$

5.2 HOL Code

```
val constructiveDilemmaRule=

 TAC_PROOF (
([], ``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
REPEAT STRIP_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC []
);
```

5.3 Session Transcript

```
> # # # # # # # # # # val constructiveDilemmaRule =                           1
    []
|- !(p :bool) (q :bool) (r :bool) (s :bool).
     (p ==> q) /\ (r ==> s) ==> p \/ r ==> q \/ s:
   thm
```

5.4 Explain Result

In 9.5.2, all of our theorem and theory have passed by HOL.

Chapter 6

Excersice 10.4.1

6.1 Problem Statement

For 10.4.1 we need to prove the therom:

$\vdash M\ \ s$

6.2 HOL Code

In 10.4.1, our relative HOL code is:

```
val problemOnethm=
TAC_PROOF(
([ ‘‘ !x: 'a.P(x) ==> M(x) ‘‘, ‘‘(P: 'a->bool) (s: 'a)‘‘],
‘‘(M:'a->bool) (s: 'a)‘‘),
RES_TAC
);
```

6.3 Session Transcript

```
> > > > # # # # # val problemOnethm =
    [..] |- M s: thm
```

6.4 Explain Result

All tests from 10.4.1 have been passed in HOL

Chapter 7

Excersice 10.4.2

7.1 Problem Statement

For 10.4.2 we need to prove the therom:

$\vdash p \Rightarrow \neg q$

7.2 HOL Code

In 10.4.2, our relative HOL code is:

```
val  problemTwothm=
TAC_PROOF(
([``p /\ q ==> r``, ``r ==> s``, ``~s``], ``p ==> ~q``),
(PAT_ASSUM ``r  ==>s``
              (fn th =>
              ASSUME_TAC
                 (DISJ_IMP (ONCE_REWRITE_RULE [DISJ_SYM] (IMP_ELIM th) )
                 )
              )
) THEN

(PAT_ASSUM ``p /\ q ==> r``
             (fn th2 =>
             ASSUME_TAC
             (DISJ_IMP (ONCE_REWRITE_RULE [DISJ_SYM] (IMP_ELIM th2))))) THEN
REPEAT STRIP_TAC THEN
RES_TAC
)
```

7.3 Session Transcript

```
> > > # # # # # # # # # # # # # # # # # val problemTwothm =
    [...] |- p  q:
   thm
```

1

7.4 Explain Result

All tests from 10.4.2 have been passed in HOL

Chapter 8

Appendix A: source code for 9.5.1, 9.5.2, and 9.5.3

The following code is from *exercise9Script*

```
(* Author: Xiaozhi Li *)
(*Project 6**)

structure exercise9Script =struct

open HolKernel Parse boolLib bossLib;

val _=new_theory "exercise9";

(* 9-5-1 *)

val absorptionRule=
TAC_PROOF (
([],``!p q. (p ==> q) ==> p ==> p/\q``),
(REPEAT STRIP_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC) );

val _=save_thm("absorptionRule",absorptionRule);
val _=export_theory();

(* 9-5-2 *)
val constructiveDilemmaRule =
 TAC_PROOF (
([], ``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
REPEAT STRIP_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC [] THEN
RES_TAC THEN
ASM_REWRITE_TAC []
);

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


(* 9-5-3 *)

val absorptionRule2=
 TAC_PROOF (
([], ``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
PROVE_TAC []
);

val _=save_thm(”absorptionRule2”,absorptionRule2);
val constructiveDilemmaRule2=
 TAC_PROOF (
([],``!p q r s.(p ==> q) /\ (r ==> s) ==> (p\/r) ==> (q\/s)``),
PROVE_TAC []
);

val _=save_thm(”constructiveDilemmaRule2”,constructiveDilemmaRule2);
val _=export_theory();
end (* structure *)
```

Chapter 9

Appendix B: source code for 10.4.1, 10.4.2

The following code is from *exercise10Script*

```
(************************************************************************)
(* Author: Xiaozhi Li                                                  *)
(*Proj 6                **)

structure exercise10Script =struct

open HolKernel Parse boolLib bossLib;

val _=new_theory "exercise10";

(*10.4.1*)

val problemOnethm=
TAC_PROOF(
([ `` !x: 'a.P(x) ==> M(x) ``, ``(P: 'a->bool) (s: 'a)``],
``(M:'a->bool) (s: 'a)``),
RES_TAC
);

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

(*        *)
(* 10.4.2 *)
val _=export_theory();



val  problemTwothm=
TAC_PROOF(
([``p /\ q ==> r``, ``r ==> s``, ``~s``], ``p ==> ~q``),
(PAT_ASSUM ``r  ==>s``
             (fn th =>
             ASSUME_TAC
                (DISJ_IMP (ONCE_REWRITE_RULE [DISJ_SYM] (IMP_ELIM th) )
                )
             )
) THEN
(PAT_ASSUM ``p /\ q ==> r``
            (fn th2 =>
            ASSUME_TAC
            (DISJ_IMP (ONCE_REWRITE_RULE [DISJ_SYM] (IMP_ELIM th2))))) THEN
REPEAT STRIP_TAC THEN
RES_TAC
)
val _=save_thm("problemTwothm",problemTwothm);

(* *)
(* 10.4.3 was not solved *)
val _=export_theory();
end (*struct*)
```