

Project 3B

Bharath Karumudi

July 26, 2019

Abstract

This project is to demonstrate the capabilities of implementing constructing and deconstructing HOL Terms using the tools and techniques - L^AT_EX, AcuTeX, emacs and ML.

Each chapter documents the given problems with a structure of:

1. Problem Statement
2. Relevant Code
3. Execution Transcripts
4. Explanation of results

Acknowledgments: Professor Marvine Hamner and Professor Shiu-Kai Chin who taught the Certified Security By Design.

Contents

1	Executive Summary	3
2	Exercise 8.4.1	4
2.1	Problem Statement	4
2.2	Relevant Code	4
2.3	Execution Transcripts	4
2.3.1	Explanation of Results	4
3	Exercise 8.4.2	5
3.1	Problem Statement	5
3.2	Relevant Code	5
3.3	Execution Transcripts	6
3.3.1	Explanation of Results	6
4	Exercise 8.4.3	7
4.1	Problem Statement	7
4.2	Relevant Code	7
4.3	Execution Transcripts	8
4.3.1	Explanation of Results	8
5	Appendix A: Chapter 8	9

Executive Summary

All the requirements for this project are statisfied specifically, and by using HOL proved the below theorems:

[conjSymThm]

$$\vdash p \wedge q \iff q \wedge p$$

[conjSymThmA11]

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

[problem1Thm]

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

Exercise 8.4.1

2.1 Problem Statement

In this exercise we need to prove the theorem: $\vdash p \Rightarrow (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \Rightarrow r$

2.2 Relevant Code

```
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);
```

2.3 Execution Transcripts

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

2.3.1 Explanation of Results

The above results shows that theorem is proved.

Exercise 8.4.2

3.1 Problem Statement

In this exercise we need to prove the theorem: $\vdash \forall p\ q. p \wedge q \iff q \wedge p$

3.2 Relevant Code

```
val conj1Thm =
let
  val th1 = ASSUME ‘‘p /\ q’’
  val th2 = CONJUNCT1 th1
  val th3 = CONJUNCT2 th1
  val th4 = CONJ th3 th2
in
  DISCH (hd(hyp th1)) th4
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_ANTISYMRULE conj1Thm conj2Thm;

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

3.3 Execution Transcripts

<pre> 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 ----- > > > # # # # # ** 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 *** > </pre>	1
--	---

3.3.1 Explanation of Results

The above results shows that theorem is proved.

Exercise 8.4.3

4.1 Problem Statement

In this exercise we need to prove the theorem: $\vdash p \Rightarrow (p \Rightarrow q) \Rightarrow (q \Rightarrow r) \Rightarrow r$

4.2 Relevant Code

```
val conj1Thm =
let
val th1 = ASSUME ‘‘p /\ q’’
val th2 = CONJUNCT1 th1
val th3 = CONJUNCT2 th1
val th4 = CONJ th3 th2
in
DISCH (hd(hyp th1)) th4
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_ANTISYMRULE conj1Thm conj2Thm;

val conjSymThmAll = GENL[‘‘p:bool’’, ‘‘q:bool’’] conjSymThm;

val _ = save_thm("conjSymThmAll", conjSymThmAll)
```

4.3 Execution Transcripts

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
-----
> > > # # # # # ** 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
> > val conjSymThmAll =
    [] |- !(p :bool) (q :bool). p /\ q <=> q /\ p:
    thm
> > # >
*** Emacs/HOL command completed ***
>

```

4.3.1 Explanation of Results

The above results shows that theorem is proved.

Appendix A: Chapter 8

The following code is from the file project3bScript.sml

```
(***** *)
(* Exercise: Chapter 8 *)
(* Author: Bharath Karumudi *)
(* Date: Jul 26, 2019 *)
(***** *)

structure project3bScript = struct
open HolKernel Parse boolLib bossLib;

val _ = new_theory "project3b";

(* ***** *)
(* Exercise: 8.4.1 *)
(* val problem1Thm = *)
(* [ ] |- p ==> (p ==> q) ==> (q ==> r) ==> r *)
(* : thm *)
(* ***** *)

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);

(* ***** *)
(* Exercise: 8.4.2 *)
(* val conjSymThm = *)
(* [ ] |- p ^ q <=> q ^ p *)
(* : thm *)
(* ***** *)
(* ***** *)
```

```

(*****)

val conj1Thm =
let
  val th1 = ASSUME ‘‘p /\ q’’
  val th2 = CONJUNCT1 th1
  val th3 = CONJUNCT2 th1
  val th4 = CONJ th3 th2
in
  DISCH (hd(hyp th1)) th4
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);

(*****
(* Exercise: 8.4.3 *)
(* val conjSymThmAll = *)
(* [] |- !p q. p /\ q <=> q /\ p *)
(* : thm *)
(* *)
(*****)

val conj1Thm =
let
  val th1 = ASSUME ‘‘p /\ q’’
  val th2 = CONJUNCT1 th1
  val th3 = CONJUNCT2 th1
  val th4 = CONJ th3 th2
in
  DISCH (hd(hyp th1)) th4
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_ANTISYMRULE conj1Thm conj2Thm;
```

```
val conjSymThmAll = GENL[‘‘p:bool ‘‘, ‘‘q:bool ‘‘] conjSymThm;
```

```
val _ = save_thm("conjSymThmAll", conjSymThmAll)
```

```
(*****
(* Exporting Theory *)
*****)
```

```
val _ = export_theory();
```

```
end (* Structure *)
```