

Project 5

Jinhao Wei

19 February 2019

Abstract

This report is a summary on my solutions of problem 8.4.1, 8.4.2 and 8.4.3. In these problems, we used *HOL* to prove theorems and used *EmitTex* to do pretty prints. Our works are contained in two folders respectively named *LaTeX* and *HOL*. In folder *LaTeX*, we provided files that are used to produce this project report. In folder *HOL*, we put all the proofs into one file named *AllScript.sml* and used some other helper files to generate HOL reports named *AllReport.pdf*.

Acknowledgments: This project follows the format and structure of *sampleTheory* provided by Professor Shiu-Kai Chin. Specifically, this report is basically a modification on the file *exampleEmitTeXReport.tex*, I merely saturated it up with my works.

Contents

1 Executive Summary	3
2 Proof of problem1Thm	4
2.1 Problem Statement	4
2.2 HOL Code Proving problem1Thm	4
2.3 Session Transcript	5
3 Proof of conjSymThm	6
3.1 Problem Statement	6
3.2 HOL Code Proving conjSymThm	6
3.3 Session Transcript	7
4 Proof of conjSymThmAll	8
4.1 Problem Statement	8
4.2 HOL Code Proving conjSymThmAll	8
4.3 Session Transcript	8
A Source Code for example1Script	9

Chapter 1

Executive Summary

All requirements for this project are satisfied. In particular, we proved all the theorems in this project, pretty printed the HOL theories, and made use of the *EmitTeX* structure to typeset HOL theorems in this report.

The following theorems are proved and their corresponding L^AT_EX macros used in this report.

[conjSymThm]

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

[conjSymThmAll]

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

[problem1Thm]

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

However, due to my incompetence on pretty print techniques, the theorems' output order is not consistent with the theorems' proof order in the report.

Chapter 2

Proof of problem1Thm

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 Proving problem1Thm

```

val problem1Thm =
let
  val th1 = ASSUME ``p:bool``
  val th2 = ASSUME ``(p:bool)==>(q:bool)``
  val th3 = ASSUME ``(q:bool) ==> (r:bool)``
  val th4 = MP th2 th1
  val th5 = IMP_TRANS th2 th3
  val th6 = MP th5 th1
  val th7 = DISCH(last(hyp(th6))) th6
  val th8 = DISCH(last(hyp(th7))) th7
in
  DISCH(last(hyp(th8))) th8
end;

```

2.3 Session Transcript

```
> > > # # # # # # # # ** types trace now on
> # # # # # # # # ** Unicode trace now off
> val problem1Thm =
let
  val th1 = ASSUME ``p:bool``
  val th2 = ASSUME ``(p:bool) ==> (q:bool)``
  val th3 = ASSUME ``(q:bool) ==> (r:bool)``
  val th4 = MP th2 th1
  val th5 = IMP_TRANS th2 th3
  val th6 = MP th5 th1
  val th7 = DISCH(last(hyp(th6))) th6
  val th8 = DISCH(last(hyp(th7))) th7
in
  DISCH(last(hyp(th8))) th8
end;
# # # # # # # # # # # # # # val problem1Thm =
  |- (p :bool) ==> (p ==> (q :bool)) ==> (q ==> (r :bool)) ==> r:
  thm
>
```

1

Chapter 3

Proof of conjSymThm

3.1 Problem Statement

Our task is to prove the theorem

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

using IMP_ANTISYM_RULE.

3.2 HOL Code Proving conjSymThm

```

val conjSymThm =
let
  val tmp1Thm =
    let
      val thm1 = ASSUME ``p/\q``
      val thm2 = CONJUNCT1 thm1
      val thm3 = CONJUNCT2 thm1
      val thm4 = CONJ thm3 thm2
      in
        DISCH ``p/\q`` thm4
      end;
  val tmp2Thm =
    let
      val thm1 = ASSUME ``q/\p``
      val thm2 = CONJUNCT1 thm1
      val thm3 = CONJUNCT2 thm1
      val thm4 = CONJ thm3 thm2
      in
        DISCH ``q/\p`` thm4
      end;
in
  IMP_ANTISYM_RULE tmp1Thm tmp2Thm
end;

```

3.3 Session Transcript

```

> val conjSymThm =
let
  val tmp1Thm =
    let
      val thm1 = ASSUME ``p/\q``
      val thm2 = CONJUNCT1 thm1
      val thm3 = CONJUNCT2 thm1
      val thm4 = CONJ thm3 thm2
      in
        DISCH ``p/\q`` thm4
      end;
  val tmp2Thm =
    let
      val thm1 = ASSUME ``q/\p``
      val thm2 = CONJUNCT1 thm1
      val thm3 = CONJUNCT2 thm1
      val thm4 = CONJ thm3 thm2
      in
        DISCH ``q/\p`` thm4
      end;
  in
    IMP_ANTISYM_RULE tmp1Thm tmp2Thm
  end;
# # # # # # # # # # # # # # # # # # # # # # val conjSymThm =
  |- (p :bool) /\ (q :bool) <=> q /\ p:
  thm

```

Chapter 4

Proof of conjSymThmAll

4.1 Problem Statement

Our task is to prove the following theorem using GENL.

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

4.2 HOL Code Proving conjSymThmAll

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

1

4.3 Session Transcript

```
> val conjSymThmAll = GENL [ ``p:bool``, ``q:bool``] conjSymThm;
val conjSymThmAll =
  |- !(p :bool) (q :bool). p /\ q <=> q /\ p:
  thm
```

1

Appendix A

Source Code for example1Script

The following code is from *AllScript.sml*, which is located in a different subdirectory than this file.

```

structure AllScript = struct

open HolKernel Parse boolLib bossLib;

val _ = new_theory "All";

(* ****
(*
(*      From now on we work on 8.4.1
(*
(* ****

val problem1Thm =
let
  val th1 = ASSUME‘‘p:bool’‘
  val th2 = ASSUME‘‘(p:bool)==>(q:bool)’‘
  val th3 = ASSUME‘‘(q:bool) ==> (r:bool)’‘

  val th4 = MP th2 th1
  val th5 = IMP_TRANS th2 th3

  val th6 = MP th5 th1

  val th7 = DISCH(last(hyp(th6))) th6
  val th8 = DISCH(last(hyp(th7))) th7
in
  DISCH(last(hyp(th8))) th8
end;

(* ****
(*      From now on we work on 8.4.2
(*
(* ****

val conjSymThm =
let

```

```

val tmp1Thm =
  let
    val thm1 = ASSUME ``p/\q``
    val thm2 = CONJUNCT1 thm1
    val thm3 = CONJUNCT2 thm1
    val thm4 = CONJ thm3 thm2
  in
    DISCH ``p/\q`` thm4
  end;
val tmp2Thm =
  let
    val thm1 = ASSUME ``q/\p``
    val thm2 = CONJUNCT1 thm1
    val thm3 = CONJUNCT2 thm1
    val thm4 = CONJ thm3 thm2
  in
    DISCH ``q/\p`` thm4
  end;
in
  IMP_ANTISYM_RULE tmp1Thm tmp2Thm
end;

(* ****
(*      From now on we work on 8.4.3
(*
(*
(* ****
val conjSymThmAll = GENL [ ``p:bool``, ``q:bool`` ] conjSymThm;

val _ = save_thm ("problem1Thm", problem1Thm);
val _ = save_thm ("conjSymThm", conjSymThm);
val _ = save_thm ("conjSymThmAll", conjSymThmAll);

val _ = export_theory();

end

```