Project 1 Report

Kyle Peppe

January 26, 2020

Abstract

This project built on our initial learning of the ML programming language and some more complicated problems that we had to solve. This project finished up with us performing some basic HOL instructions. Then I put the findings, code and output into Latex to further enhance my knowledge on making these reports. I did the exercises from the PDF book from the class 8.4.1, 8.4.2, and 8.4.3. Below are the sections that are in this report for each problem:

- 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

.

Contents

Acknowledgments: I would like to acknowledge the 2 professors, Professor Chin and Professor Hamner, that have helped me begin to understand this new ML programming language. Also to Syracuse University for accepting me to this Masters program in Cybersecurity.

Chapter 1

Executive Summary

All requirements for this project are satisfied. Specifically,

Report Contents

Our report has the following content:

Reproducibility in ML and LaTeX

The ML and LaTeX source files compile with no errors.

Chapter 2

Excercise 8.4.1

2.1 Problem statement

Prove the following theorem

```
> val problem1Thm =
        [] |- p ==> (p ==> q) ==> (q ==> r) ==> r
        : thm
```

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 t1 = hd(hyp th1)
        val t2 = hd(hyp th2)
        val t3 = hd(hyp th3)
        val th6 = DISCH t3 th4
        val th7 = DISCH t2 th6
in
        DISCH t3 th7
end

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

2.3 Session Transcript

```
---------------------------------------------------------------------
       HOL-4 [Kananaskis 11 (stdknl, built Sat Aug 19 09:30:06 2017)]        1

       For introductory HOL help, type: help "hol";
       To exit type <Control>-D
---------------------------------------------------------------------
> > > > # # # # # # # # # ** types trace now on
> # # # # # # # # # ** Unicode trace now off
> 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 t1 = hd(hyp th1)
val t2 = hd(hyp th2)
val t3 = hd(hyp th3)
val th6 = DISCH t3 th4
val th7 = DISCH t2 th6
in
DISCH t3 th7
end;
# # # # # # # # # # # # # # val problem1Thm =
    [.]
|- ((q :bool) ==> (r :bool)) ==> ((p :bool) ==> q) ==> (q ==> r) ==> q:
   thm
>
```

Chapter 3

Excercise 8.4.2

3.1 Problem statement

Prove the following theorem:

```
> val conjSymThm =
        [] |- p /\ q <=> q /\ p
        : thm
```

3.2 Relevant Code

```
val conjSymThm =
let
        val th1 = ASSUME ``p /\ q``
        val th2 = ASSUME ``q /\ p``
        val ab = CONJUNCT1 th1
        val cd = CONJUNCT2 th1
        val ba = CONJUNCT2 th2
        val bc = CONJUNCT1 th2
        val th3 = CONJ cd ab
        val th4 = CONJ ba bc
        val t1 = hd(hyp th1)
        val t2 = hd(hyp th2)
        val th5 = DISCH t1 th3
        val th6 = DISCH t2 th4
in
        IMP_ANTISYM_RULE th5 th6
end

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

3.3 Session Transcript

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
---------------------------------------------------------------------
> > > > # # # # # # # # # # ** types trace now on
> # # # # # # # # # ** Unicode trace now off
> val conjSymThm =
let
val th1 = ASSUME ``p /\ q``
val th2 = ASSUME ``q /\ p``
val ab = CONJUNCT1 th1
val cd = CONJUNCT2 th1
val ba = CONJUNCT2 th2
val bc = CONJUNCT1 th2
val th3 = CONJ cd ab
val th4 = CONJ ba bc
val t1 = hd(hyp th1)
val t2 = hd(hyp th2)
val th5 = DISCH t1 th3
val th6 = DISCH t2 th4
in
IMP_ANTISYM_RULE th5 th6
end;
# # # # # # # # # # # # # # # # val conjSymThm =
   |- (p :bool) /\ (q :bool) <=> q /\ p:
   thm
>
```

Chapter 4

Excercise 8.4.3

4.1 Problem statement

Extend your proof in Problem 2 by one step and prove:

```
> val conjSymThmAll =
       [] |- !p q. p /\ q <=> q /\ p
       : thm
```

4.2 Relevant Code

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

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

4.3 Test Case

```
---------------------------------------------------------------------
       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
> # # # # # # # # # ** Unicode trace now off
> val conjSymThm =
let
val th1 = ASSUME ``p /\ q``
val th2 = ASSUME ``q /\ p``
val ab = CONJUNCT1 th1
val cd = CONJUNCT2 th1
val ba = CONJUNCT2 th2
val bc = CONJUNCT1 th2
val th3 = CONJ cd ab
val th4 = CONJ ba bc
val t1 = hd(hyp th1)
val t2 = hd(hyp th2)
val th5 = DISCH t1 th3
val th6 = DISCH t2 th4
in
IMP_ANTISYM_RULE th5 th6
end;
# # # # # # # # # # # # # # # # val conjSymThm =
   |- (p :bool) /\ (q :bool) <=> q /\ p:
   thm
> val conjSymThmAll = GENL [``p:bool``, ``q:bool``] conjSymThm;

val _ = save_thm("conjSymThmAll", conjSymThmAll);
val conjSymThmAll =
   |- !(p :bool) (q :bool). p /\ q <=> q /\ p:
   thm
> > >
```

3

Appendix A

Source code

```
(*******************************************************************************)
(* Author: Kyle Peppe                                                          *)
(* Exercises 8.4.1, 8.4.2, and 8.4.3                                           *)
(* Date: 1/22/20                                                               *)
(*******************************************************************************)

(* Opening code needed for new theory file        *)
structure chapter8Script = struct
open HolKernel Parse boolLib bossLib;
val _ = new_theory "chapter8";


(* Exercise 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 t1 = hd(hyp th1)
        val t2 = hd(hyp th2)
        val t3 = hd(hyp th3)
        val th6 = DISCH t3 th4
        val th7 = DISCH t2 th6
in
        DISCH t3 th7
end

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


(* Exercise 8.4.2                                 *)
val conjSymThm =
let
        val th1 = ASSUME ``p /\ q``
        val th2 = ASSUME ``q /\ p``
        val ab = CONJUNCT1 th1
        val cd = CONJUNCT2 th1
        val ba = CONJUNCT2 th2
        val bc = CONJUNCT1 th2
        val th3 = CONJ cd ab
        val th4 = CONJ ba bc
```

```
        val t1 = hd(hyp th1)
        val t2 = hd(hyp th2)
        val th5 = DISCH t1 th3
        val th6 = DISCH t2 th4
in
        IMP_ANTISYM_RULE th5 th6
end

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


(* Exercise 8.4.3                                    *)
val conjSymThmAll = GENL [``p:bool``, ``q:bool``] conjSymThm;

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

(* Export Theory                                     *)
val _ = export_theory();

end
```