Assignment 1 - COMP 3400

Khalil Van Alphen
Student Number: 100863992
Email: khalil.vanalphen@scs.carleton.ca

1. Produce a knowledge base in PROPOSITIONAL LOGIC capturing the following information as closely as possible as stated here: *If Tweety is a bird and it is not an abnormal bird, then it flies. A bird is abnormal exactly when it is an ostrich, a penguin, or not an abnormal wooden bird. A wooden bird is abnormal exactly when operated under remote control. Tweety is a bird, it is not an ostrich nor a penguin nor a remote controlled wooden bird.* Prove using Prover9 that *Tweety flies.* Notice that the two forms of abnormality mentioned here may be different, one is if birds, the other is for wooden birds.

Solution: We will use the propositional variables as listed below:

a: *Tweety is a bird*

b: *Tweety flies*

c: *A bird is abnormal*

d: *A bird is an ostrich*

e: *A bird is a penguin*

f: *A bird is an abnormal wooden bird*

g: *A bird is operated by remote control*

h: *A bird is wooden*

From this we are able to deduce a the following propositional formulas:

φ_0: $(a \wedge \neg c) \rightarrow b.$

Meaning: "If Tweety is a bird and it is not an abnormal bird, then Tweety flies."

φ_1: $(d \vee e \vee \neg f) \rightarrow c$

Meaning: "If a bird is an ostrich, a penguin or not an abnormal wooden bird, it is abnormal."

φ_2: $(h \wedge g) \rightarrow f$

Meaning: "If a bird is wooden and remote controlled, it is an abnormal wooden bird."

φ_3: a

Meaning: "Tweety is a bird."

φ_4: $\neg d$

Meaning: “Tweety is not an ostrich.”

φ_5: $\neg e$

Meaning: “Tweety is not a penguin.”

φ_6: $\neg h$

Meaning: “Tweety is not wooden.”

φ_7: $\neg g$

Meaning: “Tweety is not remote controlled.”

We are attempting to prove that b is a consequence of $\varphi_0, \ldots, \varphi_7$. Prover9 will accomplish this via transforming our formulas into clauses and assumptions. Using prepositional logical equivalence, it will proceed to recursively substitute terms with more atomic parts in order to get a singular truth value. It will attempt to find a counterexample for the set of formula, and will fail.

The main section of the attached input file is as follows:

```
formulas(assumptions).

(a & -c) -> b.
(d | e | f) <-> c.
(h & g) <-> f.
a.
-d.
-e.
-h.
-g.

end_of_list.

formulas(goals).

b.

end_of_list.
```

The main section of the attached output file is as follows:

```
% Proof 1 at 0.03 (+ 0.00) seconds.
% Length of proof is 15.
% Level of proof is 4.
% Maximum clause weight is 4.
% Given clauses 0.

1 a & -c -> b # label(non_clause).  [assumption].
2 d | e | f <-> c # label(non_clause).  [assumption].
3 h & g <-> f # label(non_clause).  [assumption].
```

```
4 b # label(non_clause) # label(goal).  [goal].
5 -a | c | b.  [clausify(1)].
9 d | e | f | -c.  [clausify(2)].
11 h | -f.  [clausify(3)].
13 a.  [assumption].
14 -d.  [assumption].
15 -e.  [assumption].
16 -h.  [assumption].
18 -b.  [deny(4)].
19 c.  [back_unit_del(5),unit_del(a,13),unit_del(c,18)].
20 f.  [back_unit_del(9),unit_del(a,14),unit_del(b,15),unit_del(d,19)].
21 $F.  [back_unit_del(11),unit_del(a,16),unit_del(b,20)].
```

As is clearly seen, the simple clausifications done in steps 5 through 11 make it simple to substitute the values of the variables into the simpler formulas. By taking the assumptions from 13 through 18 and the attempt to deny at 19, the proof is only a matter of evaluating the clauses using logical operands.

2. Prove with Prover9 the theorem about idempotency in Boolean algebras shown in Chapter 2. You need to use predicate logic with ad hoc symbols for the operations and special constants involved. Make sure Prover9 understands equality. Instructions as for the previous problem.

More precisely, prove that *"For all* a*:* $a \sqcup a = a$*"* from the following axioms:

BA1 Commutativity For all a, b:

$$a \sqcup b = b \sqcup a \qquad a \sqcap b = b \sqcap a$$

BA2 Associativity: For all a, b, c:

$$(a \sqcup b) \sqcup c = a \sqcup (b \sqcup c)$$
$$(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c)$$

BA3 Distributivity: For all a, b, c:

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$$
$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$$

BA4 Neutral elements: For all a:

$$a \sqcup \hat{0} = a \qquad a \sqcap \hat{1} = a$$

BA5 "Complement": For all a:

$$a \sqcup a' = \hat{1} \qquad a \sqcap a' = \hat{0}$$

Prover9 input is as follows:

```
formulas(assumptions).

(all a all b (a | b <-> b | a)).
(all a all b (a & b <-> b & a)).
(all a all b all c ((a | b) | c <-> a | (b | c))).
(all a all b all c ((a & b) & c <-> a & (b & c))).
(all a all b all c (a | (b & c) <-> (a | b) & (a | c))).
(all a all b all c (a & (b | c) <-> (a & b) | (a & c))).
(all a (a | $F <-> a)).
(all a (a & $T <-> a)).
(all a (a | -a <-> $T)).
(all a (a & -a <-> $F)).
end_of_list.

formulas(goals).

(all a (a | a <-> a)).

end_of_list.
```

Prover9 output is as follows:

```
% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.03) seconds.
% Length of proof is 2.
% Level of proof is 1.
% Maximum clause weight is 0.
% Given clauses 0.

11 (all a (a | a <-> a)) # label(non_clause) # label(goal).  [goal].
12 $F.  [deny(11)].
```

This proof is fairly straightforward (outlined fully below).

- $a \sqcup \hat{0} = a$

  We start with a definition from our neutral element (BA4).

- $a \sqcup (a \sqcap a') = a$

  The complement rule (BA5) allows us to make a substitution.

- $(a \sqcup a) \sqcap (a \sqcup a') = a$

  The distributivity (BA3) of our algebra allows for this substitution.

- $(a \sqcup a) \sqcap \hat{1} = a$

  The complement rule (BA5) is again used to generate (1) or (TRUE).

- $(a \sqcup a) = a$

  Finally, the fact that (1) is neutral (BA4) allows it to be ignored when paired with (AND) or (UNION).

Appendix: Input/Output Files

1.INPUT:

```
set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_weight, 25).
  set(restrict_denials).
  assign(new_constants, 1).
  assign(max_seconds, 60).
end_if.

if(Mace4).   % Options for Mace4
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

(a & -c) -> b.
(d | e | f) <-> c.
(h & g) <-> f.
a.
-d.
-e.
-h.
-g.

end_of_list.

formulas(goals).

b.

end_of_list.
```

1.OUTPUT:

```
============================== prooftrans ============================
Prover9 (32) version Dec-2007, Dec 2007.
Process 232496 was started by Khalil on Khalil-PC,
Thu Feb  2 01:33:39 2017
The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".
============================== end of head ===========================

============================== end of input ==========================

============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.01 (+ 0.00) seconds.
% Length of proof is 15.
% Level of proof is 4.
% Maximum clause weight is 4.
% Given clauses 0.

1 a & -c -> b # label(non_clause).  [assumption].
```

```
2 d | e | f <-> c # label(non_clause).  [assumption].
3 h & g <-> f # label(non_clause).  [assumption].
4 b # label(non_clause) # label(goal).  [goal].
5 -a | c | b.  [clausify(1)].
9 d | e | f | -c.  [clausify(2)].
11 h | -f.  [clausify(3)].
13 a.  [assumption].
14 -d.  [assumption].
15 -e.  [assumption].
16 -h.  [assumption].
18 -b.  [deny(4)].
19 c.  [back_unit_del(5),unit_del(a,13),unit_del(c,18)].
20 f.  [back_unit_del(9),unit_del(a,14),unit_del(b,15),unit_del(d,19)].
21 $F.  [back_unit_del(11),unit_del(a,16),unit_del(b,20)].

============================== end of proof ==========================
```

2.INPUT:

```
set(ignore_option_dependencies). % GUI handles dependencies

if(Prover9). % Options for Prover9
  assign(max_weight, 25).
  set(restrict_denials).
  assign(new_constants, 1).
  assign(max_seconds, 60).
end_if.

if(Mace4).   % Options for Mace4
  assign(max_seconds, 60).
end_if.

formulas(assumptions).

(all a all b (a | b <-> b | a)).
(all a all b (a & b <-> b & a)).

(all a all b all c ((a | b) | c <-> a | (b | c))).
(all a all b all c ((a & b) & c <-> a & (b & c))).

(all a all b all c (a | (b & c) <-> (a | b) & (a | c))).
(all a all b all c (a & (b | c) <-> (a & b) | (a & c))).

(all a (a | $F <-> a)).
(all a (a & $T <-> a)).

(all a (a | -a <-> $T)).
(all a (a & -a <-> $F)).

end_of_list.

formulas(goals).

(all a (a | a <-> a)).

end_of_list.
```

2.OUTPUT:

```
============================== prooftrans ============================
Prover9 (32) version Dec-2007, Dec 2007.
Process 234724 was started by Khalil on Khalil-PC,
Thu Feb  2 02:16:52 2017
The command was "/cygdrive/c/Program Files (x86)/Prover9-Mace4/bin-win32/prover9".
============================== end of head ===========================

============================== end of input ==========================

============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.01 (+ 0.00) seconds.
% Length of proof is 2.
% Level of proof is 1.
% Maximum clause weight is 0.
% Given clauses 0.

11 (all a (a | a <-> a)) # label(non_clause) # label(goal).  [goal].
12 $F.  [deny(11)].

============================== end of proof ==========================
```