Anderson Huang
CS161

Homework #5

1. First, convert each sentence into conjunctive normal form (CNF).

(a) (Smoke => Fire) => (¬Smoke => ¬Fire)

= (¬Smoke OR Fire) => (Smoke OR ¬Fire)

= ¬(¬Smoke OR Fire) OR (Smoke OR ¬Fire)

= Smoke AND ¬Fire OR Smoke OR ¬Fire

= Smoke AND (¬Fire OR Smoke OR ¬Fire)

= Smoke AND (¬Fire OR Smoke)

There are two variables and each variable can attain two possible values. Therefore, there are four possible worlds. Let KB = knowledge base/sentence in question. A world is a model for a logical sentence if that sentence evaluates to true given the instantiations of variables in that world. A sentence is *valid* if it is true in all models. A sentence is *satisfiable* if it is true in some model.

World	Smoke	Fire	KB = Smoke AND (¬Fire OR Smoke)	Model of KB?
W1	F	F	F AND (T OR F) = F	No
W2	F	T	F AND (F or F) = F	No
W3	T	F	T AND (T OR T) = T	Yes
W4	T	T	T AND (F or T) = T	Yes

Neither. Sentence (a) is true for some worlds.

(b) (Smoke => Fire) => ((Smoke OR Heat) => Fire)

= (¬Smoke OR Fire) => (¬(Smoke OR Heat) OR Fire))

= (¬Smoke OR Fire) => (¬Smoke AND ¬Heat OR Fire)

= ¬(¬Smoke OR Fire) OR ¬Smoke AND (¬Heat OR Fire)

= Smoke AND (¬Fire OR ¬Smoke) AND (¬Heat OR Fire)

Anderson Huang
CS161

There are three variables. As such, there are eight possible worlds.

World	Smoke	Fire	Heat	KB = Smoke AND (¬Fire OR ¬Smoke) AND (¬Heat OR Fire)	Model of KB?
W1	F	F	F	F AND (T OR T) AND (T OR F) = F	No
W2	F	F	T	F AND (T OR T) AND (F OR F) = F	No
W3	F	T	F	F AND (F OR T) AND (T OR T) = F	No
W4	F	T	T	F AND (F OR T) AND (F OR T) = F	No
W5	T	F	F	T AND (T OR F) AND (T OR F) = T	Yes
W6	T	F	T	T AND (T OR F) AND (F OR F) = F	No
W7	T	T	F	T AND (F OR F) AND (T OR T) = F	No
W8	T	T	T	T AND (F OR F) AND (F OR T) = F	No

Neither. Sentence (b) is true for World 5.

(c) ((Smoke AND Heat) => Fire) ⇔ ((Smoke => Fire) OR (Heat => Fire))

= (¬(Smoke AND Heat) OR Fire) ⇔ (¬Smoke OR Fire OR ¬Heat OR Fire)

= (¬Smoke OR ¬Heat OR Fire) ⇔ (¬Smoke OR Fire OR ¬Heat)

= (¬Smoke OR ¬Heat OR Fire) => (¬Smoke OR Fire OR ¬Heat) AND
(¬Smoke OR Fire OR ¬Heat) => (¬Smoke OR ¬Heat OR Fire)

= ¬(¬Smoke OR ¬Heat OR Fire) OR (¬Smoke OR Fire OR ¬Heat) AND
¬(¬Smoke OR Fire OR ¬Heat) OR (¬Smoke OR ¬Heat OR Fire)

Let (Smoke AND Heat and ¬Fire) be a new variable X. Then

= ¬X OR X AND ¬X OR X

= TRUE and TRUE

= TRUE

Valid. Sentence (c) is always true.

Anderson Huang
CS161

World	Smoke	Fire	Heat	KB = TRUE	Model of KB?
W1	F	F	F	T	Yes
W2	F	F	T	T	Yes
W3	F	T	F	T	Yes
W4	F	T	T	T	Yes
W5	T	F	F	T	Yes
W6	T	F	T	T	Yes
W7	T	T	F	T	Yes
W8	T	T	T	T	Yes

2. (a) Define the following variables

Let M = The unicorn is mythical.
Let I = The unicorn is immortal.
Let H = The unicorn is horned.
The A = The unicorn is magical.

Then the knowledge base is as follows.

KB =
1. M => I (if the unicorn is mythical, then it is immortal.)
2. ¬M => ¬I (if the unicorn is not mythical, then it is a mortal mammal.)
3. (I OR ¬I) => H (if the unicorn is immortal or a mammal, then it is horned.)
4. H => A (If the unicorn is horned, then it is magical)

(b) We use the implication equality, (X => Y) = (¬X OR Y), to rewrite the knowledge base.

KB =
1. ¬M OR I
2. M OR ¬I
3. ¬(I OR ¬I) OR H = ¬I AND I OR H = F OR H = H
4. ¬H OR A
5. A (Resolution of sentence 3 and sentence 4)

(c) To prove facts about that the knowledge base entails, *refutation*, or proof by contradiction, will be used. This inference strategy states for a query s, KB |= S if and only if M(KB AND ¬S) = ∅. We seek to prove

i) The unicorn is mythical. (M)
ii) The unicorn is magical. (A)
iii) The unicorn is horned. (H)

i) Add ¬M to the knowledge base

6.. ¬M
7. ¬I (Resolution of sentence 2 and 6)

At this point, there are no more sentences to resolve.

The knowledge base entails that **the unicorn is magical**, and **the unicorn is horned**. However, with the current knowledge base, it is not possible to prove that the unicorn is mythical.

3. (a) P(A, B, B), P(x, y, z)
Θ = { x/A, y/A, z/A}

(b) Q(y, G(A, B)), Q(G(x, x), y)
Θ does not exist. y unifies to G(A, B) and G(x, x). This means G(A, B) = G(x, x). However x cannot unify to both A and B.

(c) Older(Father(y), y), Older(Father(x), John)
Θ = { x/John, y/John }

(d) Knows(Father(y), y), Knows(x, x)
Θ does not exist. x cannot unify to both Father(y) and y.

4. Recall first order logic is made up on relations, which describe unary or n-ary relationships between objects, predicates, which map arguments to boolean values, and functions, which are relations that uniquely map arguments to object outputs.

(a) 1. A x, Food(x) & Likes(John, x)
2. Food(Apples)
3. Food(Chicken)
4. A x, y, Eats(x, y) & ~Killed(y, x) => Food(y)
5. A x (E y, Killed(y, x)) => ~Alive(x)
6. Eats(Bill, Peanuts) & Alive(Bill)

7. A x, Eats(Bill, x) => Eats(Sue, x)

(b) Convert to CNF

1. Food(x) & Likes(John, x)
2. Food(Apples)
3. Food(Chicken)
4. ~Eats(x, y) | Killed(y, x) | Food(y)
5. A x ~(E y, ~Killed(y, x)) | ~Alive(x) = A x, y, ~Killed(y, x) | ~Alive(x)
   = ~Killed(y, x) | ~Alive(x)
6. Eats(Bill, Peanuts)
7. Alive(Bill)
8. ~Eats(Bill, x) | Eats(Sue, x)

(c) To derive new facts, use resolution

9. ~Eats(x, y) | ~Alive(x) | Food(y) [Resolution on sentence 4 and sentence 5]
10. ~Alive(Bill) | Food(Peanuts) [Resolution on 6, 9, Θ = { x/ Bill, y / Peanuts }]
11. Food(Peanuts) [Resolution on 7, 10]
12. **Likes(John, Peanuts)** [Resolution on 1, 11, Θ = { x / Peanuts }]

Therefore, John likes Peanuts.

(d) Again, use resolution.

13. Eats(Sue, Peanuts) [Resolution on sentences 6, 8, Θ = { x / Peanuts }]

Therefore, **Sue eats Peanuts.**

(e) Instead of Sentences 6 we have two new axioms. 7 stays the same.

6a*. A x, (E y, ~Eats(x, y)) => Dead(x) = A x, ~(E y, ~Eats(x, y)) | Dead(x)
= A x, y, Eats(x, y) | Dead(x) = Eats(x, y) | Dead(x)
6b*. A x, Dead(x) => ~Alive(x) = ~Dead(x) | ~Alive(x)
7. Alive(Bill)
Our new knowledge base is as follows.

1. Food(x) & Likes(John, x)
2. Food(Apples)
3. Food(Chicken)
4. ~Eats(x, y) | Killed(y, x) | Food(y)
5. ~Killed(y, x) | ~Alive(x)
6. Eats(x, y) | Dead(x)
7. ~Dead(x) | ~Alive(x)
8. Alive(Bill)

Anderson Huang
CS161

9. ~Eats(Bill, x) | Eats(Sue, x)

10. Eats(x, y) | ~Alive(x) [Resolution on Sentences 6, 7]
11. Eats(Bill, y) [Resolution on Sentences, 8, 10, Θ = { x / Bill }]
12. Killed(y, Bill) | Food(y) [Resolution on Sentences 4, 11]
13. ~Alive(Bill) | Food(y) [Resolution on Sentences 5, 12 Θ = { x / Bill }]
14. Food(y)
15. Eats(Sue, y) [Resolution on Sentences 9, 11]

Out of new facts to resolve. The only thing we can say is that Sue eats something.