HW8

Zeming Xie

Question 1:

KB: p ⇒ (q ⇒ r)

¬ ((p ∧ q) ⇒ (q ⇒ r)) // add negation of query

Change everything to CNF:

p ⇒ (q ⇒ r) // starting sentence

¬p∨(q ⇒ r) // implication elimination

¬p∨(¬q ∨ r) // implication elimination

¬p∨¬q ∨ r // move negation inwards & distribute ∨ over ∧

¬ ((p ∧ q) ⇒ (q ⇒ r)) // starting sentence

¬ (¬ (p ∧ q) ∨ (q ⇒ r)) // implication elimination

¬¬(p ∧ q) ∧¬(q ⇒ r) // de Morgan

(p ∧ q) ∧¬(q ⇒ r) // double-negation elimination

(p ∧ q) ∧¬(¬q∨r) // implication elimination

(p ∧ q) ∧(¬¬q∧¬r) // de Morgan

(p ∧ q) ∧(q∧¬r) // double-negation elimination

p ∧ q ∧ q ∧ ¬r // distribute ∨ over ∧

p ∧ q ∧ ¬r // factoring

Now the new KB with CNF is:

1: ¬p∨¬q ∨ r

2: p ∧ q ∧ ¬r

3: ¬ (p∨q) ∨ r // de Morgan to 1

4: p ∧ q // and-elimination to 3

5: ¬r // and-elimination to 3

6: r // resolve 3 and 4

7: empty // resolve 5 and 6

Therefore, according to resolution, we can derive the query:

(p ∧ q) ⇒ (q ⇒ r)

Question 2:



Child(x): x is a child Loves(x, y): x loves y

Reindeer(x): x is a reindeer Loves(x, y): x loves y

Reindeer(x): x is a reindeer RedNose(x): x has red nose

RedNose(x): x has red nose Weird(x): x is weird Clown(x): x is a clown

Clown(x): x is a clown Reindeer(x): x is a reindeer

Weird(x): x is weird Loves(x, y): x loves y

Child(x): x is a child



Austinite(x): x is an Austinite Conservative(x): x is conservative

Armadillo(x): x is an armadillo Loves(x, y): x loves y

Wear(x, y): x wears y Aggie(x): x is an aggie

Aggie(x): x is an aggie Dog(x): x is a dog Loves(x, y): x loves y

Armadillo(x): x is an armadillo Dog(x): x is a dog Loves(x, y): x loves y

Austinite(x): x is an Austinite Wear(x, y): x wears y

1. Austinite(Clem) Wear(Clem, )

Austinite(x): x is an Austinite Conservative(x): x is conservative

Question 3:

1. I(x): x is innocent F(x): x and Victor are friends L(x): x is a liar

Alice: (I(Alice) ∧ F(Barney) ∧ ¬F(Caddy)) ∨ L(Alice)

Barney: (I(Barney) ∧ ¬F(Barney)) ∨ L(Barney)

Caddy: (I(Caddy) ∧ F(Barney)) ∨ L(Caddy)

1. CNF form of (1)

(I(Alice) ∨ L(Alice)) ∧ (F(Barney) ∨ L(Alice)) ∧ (¬F(Caddy) ∨ L(Alice))

(I(Barney) ∨ L(Barney)) ∧ (¬F(Barney) ∨ L(Barney))

(I(Caddy) ∨ L(Caddy)) ∧ (F(Barney) ∨ L(Caddy))

CNF form of (2)

1. Either or or
2. F(Caddy) ∨ L(Caddy)
3. It is unsatisfiable

If then, using second sentence from part(2) we can get I(Barney) and I(Caddy)

Using third sentence from part(2) and I(Barney) and I(Caddy) we can get (Barney) and

(Caddy).

Using second and third sentence from part(1) and (Barney) and (Caddy) we can get ¬F(Barney) and F(Barney) which is empty, therefore it is not

If then, using second sentence from part(2) we can get I(Alice) and I(Caddy)

Using third sentence from part(2) and I(Alice) and I(Caddy) we can get (Alice) and

(Caddy).

Using first and third sentence from part(1) and (Alice) and (Caddy) we can get F(Barney).

Using first sentence from part(2) and F(Barney), we can get

Using the assumption and I(Barney), we get empty, which shows it is not

If then, using second sentence from part(2) we can get I(Barney) and I(Alice)

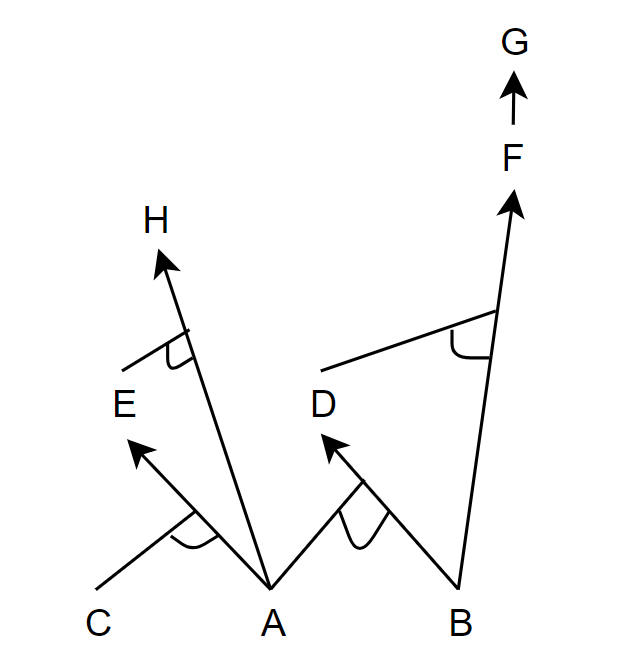
Using third sentence from part(2) and I(Barney) and I(Alice) we can get (Barney) and

(Alice).

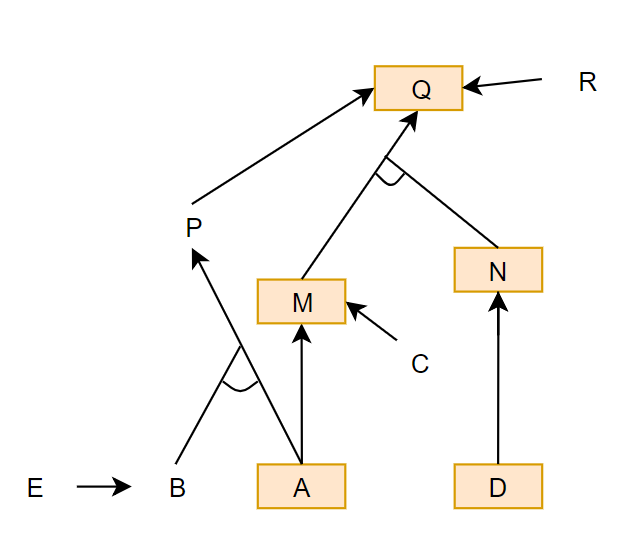
Using second and first sentence from part(1) and (Alice) and (Barney) we can get ¬F(Barney) and F(Barney) which is empty, therefore it is not

Therefore, all three possible goals are false under resolution. So, the KB is unsatisfiable.

Question 4:



H is true, A and C are true from KB, then we can get E, then from A and E, we can get H

1. 

1: P ⇒ Q

2: E ⇒ B

3: R ⇒ Q

4: M ∧ N ⇒ Q

5: A ∧ B ⇒ P

6: A ⇒ M

7: C ⇒ M

8: D ⇒ N

9: D

10: A

11: M // Modus Ponens to 6, 10

12: N // Modus Ponens to 8, 9

13: Q // Modus Ponens to 11, 12, 4

Question 5: