Naim Ayat

CS 161: Fundamentals of Artificial Intelligence

Assignment 5 – March 12, 2018

1. a) Neither.

|  |  |  |
| --- | --- | --- |
| fire | smoke | (smoke → fire) → (~smoke → ~fire) |
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

b) Neither.

|  |  |  |  |
| --- | --- | --- | --- |
| fire | heat | smoke | (smoke → fire) → ((smoke \/ heat) → fire) |
| F | F | F | T |
| F | F | T | T |
| F | T | F | F |
| F | T | T | T |
| T | F | F | T |
| T | F | T | T |
| T | T | F | T |
| T | T | T | T |

c) Valid.

|  |  |  |  |
| --- | --- | --- | --- |
| fire | heat | smoke | ((smoke /\ heat) → fire) ↔ ((smoke → fire) \/ (heat → fire)) |
| F | F | F | T |
| F | F | T | T |
| F | T | F | T |
| F | T | T | T |
| T | F | F | T |
| T | F | T | T |
| T | T | F | T |
| T | T | T | T |

1. a) Knowledge base:

*P1 = Mythical → Immortal*

*P2 = ~Mythical → ~ Immortal /\ Mammal*

*P3 = Immortal \/ Mammal → Horned*

*P4 = Horned → Magical*

b) Knowledge base converted to CNF:

* *~Mythical \/ Immortal*
* *(Mythical \/ ~Immortal) /\ (Mythical \/ Mammal)*
* *(~Immortal \/ Horned) /\ (~Mammal \/ Horned)*
* *~Horned \/ Magical*

c) It is not possible to prove that the unicorn is magical from the knowledge base.

However, it is provable that the unicorn is horned and mythical:

* + 1. *~Immortal → ~Mythical (contrapositive of P1)*
    2. *~Immortal → ~Immortal /\ Mammal (hypothetical syllogism applied to i and P2)*
    3. *Immortal \/ (~Immortal /\ Mammal) (definition of implication applied to ii)*
    4. *(Immortal \/ ~Immortal) /\ (Immortal \/ Mammal) (iii converted to CNF)*
    5. *Immortal \/ Mammal (tautological simplification of iv)*
    6. *Horned (modus ponens applied to v and P3)*
    7. *Mythical (modus ponens applied to vi and P4)*

1. a) P(A, B, B), P(x, y, z): {x/A}

P(A, B, B), P(A, y, z): {x/A, y/B}

P(A, B, B), P(A, B, z): **{x/A, y/B, z/B}**

b) Q(y, G(A, B)), Q(G(x, x), y) : {y/G(x, x)},

Q(G(x, x), G(A, B)), Q(G(x, x), G(x, x)): {y/G(x, x)}

Q(G(x, x), G(A, B)), Q(G(x, x), G(x, x)): {y/G(x, x), x/A}

Q(G(A, A), G(A, B)), Q(G(A, A), G(A, A)) : {y/G(x, x), x/A}

**A cannot be unified with B, hence no general unifier exists.**

c) Older(Father(y), y), Older(Father(x), John)

Older(Father(y), y), Older(Father(x), John): {x/y}

Older(Father(x), x), Older(Father(x), John): {y/x, x/John}

**{y/John, x/John}**

d) Knows(Father(y), y), Knows(x, x): {x/Father(y)}

Knows(Father(y), y), Knows(Father(y), Father(y)): {x/Father(y)}

**Father(y) cannot be unified with y, hence no general unifier exists.**

1. a) First-order logic:
2. *(A x) (Food(x) → Likes(John, x))*
3. *Food(Apples)*
4. *Food(Chicken)*
5. *(A x A y) (Eats(x, y) /\ ~Killed(x, y) → Food(y))*
6. *(A x A y) (Killed(x, y) → ~Alive(x))*
7. *Alive(Bill) /\ Eats(Bill, Peanuts)*
8. *(A x) (Eats(Bill, x) → Eats(Sue, x))*

b) First-order logic converted to CNF:

1. *~Food(x) \/ Likes(John, x)*
2. *Food(Apples)*
3. *Food(Chicken)*
4. *~Eats(x, y) \/ Killed(x, y) \/ Food(y)*
5. *~Killed(x, y) \/ ~Alive(x)*
6. *{ [Part A: (Alive(Bill))]  
    [Part B: (Eats(Peanuts, Bill))] }*
7. *~Eats(Bill, x) \/ Eats(Sue, x)*

c) Proof that John likes Peanuts:

*~Likes(John, Peanuts) Hypothesis*

*~Food(Peanuts) (Resolve with I)*

*~Eats(x, Peanuts) \/ Killed(x, Peanuts) (Resolve with IV)*

*Killed(Bill, Peanuts) (Resolve with VI part B)*

*~Alive(Bill) (Resolve with V)*

*N/A (Resolve with VI part A)*

Since we have proven ~Likes(John, Peanuts) is false, then it must be true that John

does, in fact, like peanuts.

d) “What food does Sue eat?” *= (E x Food(x) /\ Eats(Sue, x))*. In CNF, this is

equivalent to: *(~Food(x) \/ ~Eats(Sue, x)).*

*~Food(x) \/ ~Eats(Sue, x)*

*~Eats(Bill, x) \/ ~Food(x) (Resolve with VII)*

*~Food(Peanuts) (Resolve with VI part B)*

Now we have the same proof from part (c), starting with the second resolution.

Hence, *Eats(Peanuts, Sue)*; the unifier is *{x/Peanuts}*.

e) We replace IV with the following:

1. *(A x) (~E y Eats(x, y)) → Dead(x)*
2. *(A x) (Dead(x) → ~Alive(x))*
3. *Alive(Bill)*

Where f(x) represents an unknown food, the new axioms can be written in CNF as:

1. *Eats(x, f(x)) \/ Dead(x)*
2. *~Dead(x) \/ ~Alive(x)*
3. *Alive(Bill)*

The entire knowledge base in CNF is now:

*I. ~Food(x) \/ Likes(John, x)*

*II. Food(Apples)*

*III. Food(Chicken)*

*IV. ~Eats(x, y) \/ Killed(x, y) \/ Food(y)*

*V. ~Killed(x, y) \/ ~Alive(x)*

*VI. { [Part A: Eats(x, f(x)) \/ Dead(x)],   
 [Part B: ~Dead(x) \/ ~Alive(x)],   
 [Part C: Alive(Bill)] }*

*VII. ~Eats(Bill, x) \/ Eats(Sue, x)*

Now, we find what food Sue eats:

*~Eats(Sue, z) \/ ~Food(z) What does Sue eat?*

*~Eats(Bill, z) \/ ~Food(z) (Resolve with VII)*

*~Food(f(Bill)) \/ Dead(Bill) (Resolve with VI part A)*

*~Food(f(Bill)) \/ ~Alive(Bill) (Resolve with VI part C)*

*~Food(f(Bill)) (Resolve with VI part B)*

*~Eats(x, f(Bill)) \/ Killed(x, f(Bill)) (Resolve with IV)*

*~Eats(x, f(Bill)) \/ ~Alive(z) (Resolve with V)*

*~Eats(Bill, f(Bill)) (Resolve with VI part B)*

*~Dead(Bill) (Resolve with VI part A)*

*~Alive(Bill) (Resolve with VI part C)*

*N/A (Resolve with VI part B)*

Hence, Sue eats whatever Bill eats.