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

2. a) Knowledge base:

$P1 = Mythical \rightarrow Immortal$

$P2 = \sim Mythical \rightarrow \sim Immortal \wedge Mammal$

$P3 = Immortal \vee Mammal \rightarrow Horned$

$P4 = Horned \rightarrow Magical$

b) Knowledge base converted to CNF:

- $\sim Mythical \vee Immortal$
- $(Mythical \vee \sim Immortal) \wedge (Mythical \vee Mammal)$
- $(\sim Immortal \vee Horned) \wedge (\sim Mammal \vee Horned)$
- $\sim Horned \vee Magical$

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:

i.	$\sim Immortal \rightarrow \sim Mythical$	*(contrapositive of P1)*
ii.	$\sim Immortal \rightarrow \sim Immortal \wedge Mammal$	*(hypothetical syllogism applied to i and P2)*
iii.	$Immortal \vee (\sim Immortal \wedge Mammal)$	*(definition of implication applied to ii)*
iv.	$(Immortal \vee \sim Immortal) \wedge (Immortal \vee Mammal)$	*(iii converted to CNF)*
v.	$Immortal \vee Mammal$	*(tautological simplification of iv)*
vi.	$Horned$	*(modus ponens applied to v and P3)*
vii.	$Mythical$	*(modus ponens applied to vi and P4)*

3. 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.

4. a) First-order logic:

   i. *(A x) (Food(x) → Likes(John, x))*
   ii. *Food(Apples)*
   iii. *Food(Chicken)*
   iv. *(A x A y) (Eats(x, y) /\ ~Killed(x, y) → Food(y))*
   v. *(A x A y) (Killed(x, y) → ~Alive(x))*
   vi. *Alive(Bill) /\ Eats(Bill, Peanuts)*
   vii. *(A x) (Eats(Bill, x) → Eats(Sue, x))*

b) First-order logic converted to CNF:

   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: (Alive(Bill))]*
   [Part B: (Eats(Peanuts, Bill))] }
   VII. *~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:

A. *(A x) (~E y Eats(x, y)) → Dead(x)*
B. *(A x) (Dead(x) → ~Alive(x))*
C. *Alive(Bill)*

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

A. *Eats(x, f(x)) \/ Dead(x)*
B. *~Dead(x) \/ ~Alive(x)*
C. *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.