Xiaojia Quan
CS 161
HW6

1.

(a) P(A, B, C), P(x, y, z)

Unifier: {x/A, y/B, z/C}

(b) Q(y, G(B, A), D), Q(G(x, x), y, D).

The unifier does not exist because x cannot bind to both A and B

(c) R(x,z,A) , R(A,z,y)

{x/A, y/A, z/B}

(d) Older(Father(y), John), Older(Father(x), x).

{x/John, y/John}

(e) Knows(y,y), Knows(Father(x),x).

The unifier does not exist since we cannot force y/Father(y)

2.

- John likes all kinds of food.
- Apples are food.
- Chicken is food.
- Anything someone eats and isn't killed by is food.
- If you are killed by something, you are not alive.
- Bill eats peanuts and is still alive. *
- Sue eats everything Bill eats.

1. (a) Translate these sentences into formulas in first-order logic.
    a. (Ax) (Food(x) => Likes(John, x))
    b. Food(Apples)
    c. Food(Chicken)
    d. ((A a b) (Eat(a, b) & (~Kill (b, a))) => Food(b))

e. ((A q r) Kills(q, r) => (~ Alive(r)))
f. Eats(Bill, Peanuts) & Alive(Bill)
g. (Az) ((Eats (Bill, z)) => Eats (Sue, z))

(b) Convert the formulas of part (a) into CNF (also called clausal form).

a. ~Food(x) | Likes(John, x)

b. Food(Apples)

c. Food(Chicken)

d. ~Eat(a, b) | Kill(b, a) | Food(b)

e. ~Kills(q, r) | ~Alive(r)

f. Eats(Bill, Peanut) & Alive(Bill)

g. ~Eats(Bill, z) | Eats(Sue, z)

(c) Prove that John likes peanuts using resolution.

a. ~Food(x) | Likes(John, x)

b. Food(Apples)

c. Food(Chicken)

d. ~Eat(a, b) | Kill(b, a) | Food(b)

e. ~Kills(q, r) | ~Alive(r)

f. Eats(Bill, Peanut) & Alive(Bill)

g. ~Eats(Bill, z) | Eats(Sue, z)

Prove: h. ~Likes(John, peanuts)

---

i. ~Food(Peanuts)	a, h
j. ~Eats(Bill, Peanuts) | Kills(Peanuts, Bill)	d, i, a/Bill, b/Peanuts
k. Eats(Bill, Peanuts)	f
l. Kills(Peanuts, Bill)	j, k

m. ~Alive(Bill) r/Bill, e, l

n. Alive(Bill) f

o. False

contradiction between m and n. Thus, ~Like(John, peanuts) is false. Likes(John, peanuts) has been proved.

(d) Use resolution to answer the question, “What food does Sue eat?”

a. ~Food(x) | Likes(John, x)

b. Food(Apples)

c. Food(Chicken)

d. ~Eat(a, b) | Kill(b, a) | Food(b)

e. ~Kills(q, r) | ~Alive(r)

f. Eats(Bill, Peanut) & Alive(Bill)

g. ~Eats(Bill, z) | Eats(Sue, z)

Eats(Sue, peanut) f. g. z/Peanut

(e)

(CNF)

1. Eats(x, y) | Die(x)

2. ~ Die(x) | ~Alive(x)

3. Alive(Bill)

Ans:

a. ~Food(x) | Likes(John, x)

b. Food(Apples)

c. Food(Chicken)

d. ~Eat(a, b) | Kill(b, a) | Food(b)

e. ~Kills(q, r) | ~Alive(r)

g. ~Eats(Bill, z) | Eats(Sue, z)

1. Eats(o, q) | Die(o)

2. ~ Die(w) | ~Alive(w)

3. Alive(Bill)

---

4. ~Die(Bill)	x/Bill
5. Eats(Bill, q)	o/Bill
6.Eats(Bill， M)	q/M
7. ~Eats(Bill, M) | Eats(Sue, M)	z/M
8. Eats(Sue， M)	6， 7