Bradley Mont
CS 161
Disc 1C
804-993-030

Homework 6

1.

a. $P(A, A, B)$ $P(x, y, z)$

$\theta = \{x/A, y/A, z/B\}$

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

can't be unified b/c can't bind x to A and B

c. $R(x, A, z)$ $R(B, y, z)$

$\theta = \{x/B, y/A\}$

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

$\theta = \{x/John, y/John\}$

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

no valid unifier because that would imply y is its own father, which does not make sense

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.

a. First-Order sentences:

- $\forall x \; Food(x) \Rightarrow Likes(John, x)$
- $Food(Apples)$
- $Food(Chicken)$
- $\forall x \; (\exists y \; Eats(y,x) \land \neg Kills(x,y)) \Rightarrow Food(x)$
- $\forall x \; (\exists y \; Kills(y,x)) \Rightarrow \neg Alive(x)$
- $Eats(Bill, Peanuts) \land Alive(Bill)$
- $\forall x \; Eats(Bill, x) \Rightarrow Eats(Sue, x)$

6. convert to CNF

1. eliminate =>, <=> :

- $\forall x\ \neg Food(x) \vee Likes(John, x)$
- $Food(Apples)$
- $Food(Chicken)$
- $\forall x\ \neg(\exists y\ Eats(y,x) \wedge \neg Kills(x,y)) \vee Food(x)$
- $\forall x\ \neg(\exists y\ Kills(y,x)) \vee \neg Alive(x)$
- $Eats(Bill, Peanuts) \wedge Alive(Bill)$
- $\forall x\ \neg Eats(Bill, x) \vee Eats(Sue, x)$

2. move $\neg$ inwards

- $\forall x\ \neg Food(x) \vee Likes(John, x)$
- $Food(Apples)$
- $Food(Chicken)$
- $\forall x\ [\forall y\ \neg Eats(y,x) \vee Kills(x,y)] \vee Food(x)$
- $\forall x\ [\forall y\ \neg Kills(y,x)] \vee \neg Alive(x)$
- $Eats(Bill, Peanuts) \wedge Alive(Bill)$
- $\forall x\ \neg Eats(Bill, x) \vee Eats(Sue, x)$

3. standardize variables:

- $\forall a\ \neg Food(a) \vee Likes(John, a)$
- $Food(Apples)$
- $Food(Chicken)$
- $\forall b\ [\forall c\ \neg Eats(c,b) \vee Kills(b,c)] \vee Food(b)$
- $\forall d\ [\forall e\ \neg Kills(e,d)] \vee \neg Alive(d)$
- $Eats(Bill, Peanuts) \wedge Alive(Bill)$
- $\forall f\ \neg Eats(Bill, f) \vee Eats(Sue, f)$

4. skolemize - there aren't any $\exists$

5. drop universal quantifiers

- $\neg Food(a) \vee Likes(John, a)$
- $Food(Apples)$
- $Food(Chicken)$
- $\neg Eats(c,b) \vee Kills(b,c) \vee Food(b)$
- $\neg Kills(e,d) \vee \neg Alive(d)$
- $Eats(Bill, Peanuts) \wedge Alive(Bill)$
- $\neg Eats(Bill, f) \vee Eats(Sue, f)$

6. distribute - nothing to distribute

CNF clauses:

$$\left.\begin{array}{ll}
1. & \neg Food(a) \vee Likes(John, a) \\
2. & Food(Apples) \\
3. & Food(Chicken) \\
4. & \neg Eats(c, b) \vee Kills(b, c) \vee Food(b) \\
5. & \neg Kills(e, d) \vee \neg Alive(d) \\
6. & Eats(Bill, Peanuts) \\
7. & Alive(Bill) \\
8. & \neg Eats(Bill, f) \vee Eats(Sue, f)
\end{array}\right\} \Delta$$

c. Prove John likes peanuts using resolution

Δ: CNF clauses on prev. page

α: Likes(John, Peanuts)

- To prove $\Delta \models \alpha$, we prove that $\Delta \wedge \neg\alpha$ is unsatisfiable:

1.	$\neg Food(a) \lor Likes(John, a)$		
2.	$Food(Apples)$		
3.	$Food(Chicken)$		
4.	$\neg Eats(c,b) \lor Kills(b,c) \lor Food(b)$		
5.	$\neg Kills(e,d) \lor \neg Alive(d)$		
6.	$Eats(Bill, Peanuts)$		
7.	$Alive(Bill)$		
8.	$\neg Eats(Bill, f) \lor Eats(Sue, f)$		
9.	$\neg Likes(John, Peanuts)$		
10.	$\neg Kills(e, Bill)$	5,7	$\theta = \{d/Bill\}$
11.	$\neg Food(Peanuts)$	1,9	$\theta = \{a/Peanuts\}$
12.	$\neg Eats(c, Peanuts) \lor Kills(Peanuts, c)$	4,11	$\theta = \{b/Peanuts\}$
13.	$Kills(Peanuts, Bill)$	6,12	$\theta = \{c/Bill\}$
14.	empty clause (contradiction)	10,13	$\theta = \{e/Peanuts\}$

Since we found a contradiction, $\Delta \land \neg\alpha$ is unsatisfiable, so $\Delta \models \alpha$: John likes peanuts.

d. What does Sue eat?

1. $\neg Food(a) \lor Likes(John, a)$
2. $Food(Apples)$
3. $Food(Chicken)$
4. $\neg Eats(c,b) \lor Kills(b,c) \lor Food(b)$
5. $\neg Kills(e,d) \lor \neg Alive(d)$
6. $Eats(Bill, Peanuts)$
7. $Alive(Bill)$
8. $\neg Eats(Bill, f) \lor Eats(Sue, f)$
9. $Eats(Sue, Peanuts)$ 6,8 $\theta = \{f / Peanuts\}$

Through resolution, we derived the sentence $Eats(Sue, Peanuts)$. Therefore, Sue eats Peanuts.

e.

new sentences:

- If you don't eat, you die
- If you die, you are not alive
- Bill is alive

FOL:

- $\forall g \ \neg[\exists h \ Eats(g,h)] \Rightarrow Die(g)$
- $\forall i \ \ Die(i) \Rightarrow \neg Alive(i)$
- $Alive(Bill)$

convert new sentences to CNF:

1. eliminate =>

- $\forall g \ [\exists h \ Eats(g,h)] \lor Die(g)$
- $\forall i \ \ \neg Die(i) \lor \neg Alive(i)$
- $Alive(Bill)$

2. done
3. done

4. skolemize

- $\forall g \ [Eats(g, F(g))] \lor Die(g)$
- $\forall i \ \ \neg Die(i) \lor \neg Alive(i)$
- $Alive(Bill)$

5. drop universal quantifiers

- $Eats(g, F(g)) \lor Die(g)$
- $\neg Die(i) \lor \neg Alive(i)$
- $Alive(Bill)$

6. done

Now we use resolution to derive what Sue eats:

1.	$\neg Food(a) \lor Likes(John, a)$		
2.	$Food(Apples)$		
3.	$Food(Chicken)$		
4.	$\neg Eats(c, b) \lor Kills(b, c) \lor Food(b)$		
5.	$\neg Kills(e, d) \lor \neg Alive(d)$		
6.	$\neg Eats(Bill, f) \lor Eats(Sue, f)$		
7.	$Eats(g, F(g)) \lor Die(g)$		
8.	$\neg Die(i) \lor \neg Alive(i)$		
9.	$Alive(Bill)$		
10.	$\neg Die(Bill)$	8, 9	$\theta = \{i/Bill\}$
11.	$Eats(Bill, F(g))$	7, 10	$\theta = \{g/Bill\}$
12.	$Eats(Sue, F(g))$	6, 11	$\theta = \{f/F(g)\}$

Since we can't derive any more statements about what exactly Bill and Sue eat, we cannot conclude exactly what Sue eats.

#3: (Screenshot of my .txt file)

```
3.1. The SAT instance for 3-coloring graph 1 is not satisfiable.

3.2. The SAT instance for 4-coloring graph 1 is satisfiable.

3.3. The answers of these two SAT instances tell us that graph 2 can
only be k-colored where k >= 4.

A solution to coloring graph 1 is the following variable index
assignments:
-1 -2 -3 4 -5 -6 7 -8 -9 10 -11 -12 -13 -14 15 -16 17 -18 -19 -20
-21 22 -23 -24 25 -26 -27 -28

Each of these variable indices was calculated using the formula:
variable index = (n - 1) * k + c
where k = 4, n is in [1,7], and c is in [1,4]

A positive variable index indicates that a node was colored that
color, so we can derive the following results from the assignments:

4  -> node 1 was colored color #4
7  -> node 2 was colored color #3
10 -> node 3 was colored color #2
15 -> node 4 was colored color #3
17 -> node 5 was colored color #1
22 -> node 6 was colored color #2
25 -> node 7 was colored color #1

3.4. The minimum number of colors required to properly color graph 2
is 8.
```