CS 161 HW 6

1a) {x/A, y/B, z/B}

1b) not unifiable

1c) {x/B, y/A}

1d) {x/John, y/John}

1e) not unifiable

2a)

* ∀ x Food(x) => Likes(John, x)
* Food(Apples)
* Food(Chicken)
* ∀ x, y [Eats(y, x) ^ ¬Kills(x, y)] => Food(x)
* ∀ x, y [Kills(x, y) => ¬Alive(y)]
* Eats(Bill, Peanuts) ^ Alive(Bill)
* ∀ x Eats(Bill, x) => Eats(Sue, x)

2b)

* ¬Food(x1) v Likes(John, x1)
* Food(Apples)
* Food(Chicken)
* ¬Eats(y1, x2) v Kills(x2, y1) v Food(x2)
* ¬Kills(x3, y2) v ¬Alive(y2)
* Eats(Bill, Peanuts) ^ Alive(Bill)
* ¬Eats(Bill, x4) v Eats(Sue, x4)

2c)

1. ¬Likes(John, Peanuts)
2. ¬Food(x1) v Likes(John, x1)
3. Food(Apples)
4. Food(Chicken)
5. ¬Eats(y1, x2) v Kills(x2, y1) v Food(x2)
6. ¬Kills(x3, y2) v ¬Alive(y2)
7. Eats(Bill, Peanuts)
8. Alive(Bill)
9. ¬Eats(Bill, x4) v Eats(Sue, x4)
10. Kills(Peanuts, Bill) v Food(Peanuts) { y1/Bill, x2/Peanuts} (5, 7)
11. ¬Kills(x3, Bill) { y2/Bill} (6, 8)
12. Food(Peanuts) { x3/Peanuts} (10, 11)
13. Likes(John, Peanuts) { x1/Peanuts} (2, 12)
14. Contradiction (1,13)

Since there is a contradiction, John must like Peanuts.

2d)

1. ¬Food(x1) v Likes(John, x1)
2. Food(Apples)
3. Food(Chicken)
4. ¬Eats(y1, x2) v Kills(x2, y1) v Food(x2)
5. ¬Kills(x3, y2) v ¬Alive(y2)
6. Eats(Bill, Peanuts)
7. Alive(Bill)
8. ¬Eats(Bill, x4) v Eats(Sue, x4)
9. Eats(Sue, Peanuts) { x4/Peanuts} (6, 8)

By resolution, Sue eats peanuts.

2e)

1. ¬Food(x1) v Likes(John, x1)
2. Food(Apples)
3. Food(Chicken)
4. ¬Eats(y1, x2) v Kills(x2, y1) v Food(x2)
5. ¬Kills(x3, y2) v ¬Alive(y2)
6. ¬Eats(Bill, x4) v Eats(Sue, x4)
7. Eats(y3, x5) v Dies(y3)
8. ¬Dies(y4) v ¬Alive(y4)
9. Alive(Bill)
10. ¬Dies(Bill) { y4/Bill} (8, 9)
11. Eats(Bill, x5) { y3/Bill} (7, 10)
12. Eats(Sue, x4) (6, 11)

Sue can eat anything, since you can set x4 to anything.