Homework4

Qi Mao

`maoxx241@umn.edu`

April 4, 2019

1   Question1:

[20 points] For each of the following sentences, decide if the logic sentence given is a correct translation of the English sentence or not. If not explain briefly why not and correct it:

- 1. All houses have at least one bathroom.
  $\forall$ x [House(x) $\wedge$ $\exists$ y Bathroom(y)] $\Rightarrow$ In(x,y)

  Incorrect. This sentence means there is a Bathroom in every house. This should be:
  $\forall$x house(x) $\Rightarrow$ $\exists$y bathroom(y) $\wedge$ In(y, x)

- 2. There is a house in Minneapolis which costs more than any other house.
  $\forall$ x [House(x) $\wedge$ In(x,Minneapolis)] $\Rightarrow$ [$\exists$ y House(y) $\wedge$ Cost(y) > Cost(x)]

  Incorrect. This sentence means any houses in Minneapolis, there is house cost more than it. This should be:
  $\exists$y house(y) $\wedge$ In(y, Minneapolis) $\wedge$ [$\forall$x house(x) $\Rightarrow$ cost(y)>cost(x) $\wedge$ x$\neq$y]

- 3. There is only one house in Minneapolis that is pink.
  $\exists$ x House(x) $\wedge$ In(x,Minneapolis) $\wedge$ Color(x,Pink)

  Incorrect. It doesn't show that there is only one. This should be:
  $\exists$ x House(x) $\wedge$ In(x,Minneapolis) $\wedge$ Color(x,Pink) $\wedge$ $\forall$y [house(y) $\wedge$ in(y, Minneapolis) $\wedge$ Color(y,Pink) $\Rightarrow$ x=y]

- 4. Some houses cost less than some apartments.
  $\exists$ x House(x) $\wedge$ $\exists$ y Apartment(y) $\Rightarrow$ Cheaper(x,y)

  Incorrect. $\Rightarrow$should be $\wedge$. This should be:
  $\exists$ x House(x) $\wedge$ $\exists$ y Apartment(y) $\wedge$ Cheaper(x,y)

- 5. There are fastfood places in every city.
  ∀ x ∀ y [City(x) ∧ Fastfood (y)] ⇒ In(y,x)

  Incorrect. It means all fastfood places in all cities. This is also incorrect. This should be:
  ∃ x Fastfood(x) ∧ ∀ y City(y) ⇒ In(x,y)

2   Question2

[20 points] Convert these English sentences to predicate calculus, using the following predicates: Cat(x) = x is a cat; Bird(x) = x is a bird; Dog(x) = x is a dog; Animal(x) = x is an animal; Person(x) = x is a person. Owns(x,y) = x owns y; Likes(x,y) = x likes y, Kill(x,y) = x kills y. John is a constant.

- 1. Birds do not like cats.
  ∀ x,y Bird(x) ∧ Cat(y) ⇒¬Likes(x,y)

- 2. Any person who owns a cat does not own a bird.
  ∀ x[Person(x) ∧ ∃y Cat(y) ∧ Owns(x,y)] ⇒ ¬[∃z Bird(z) ∧ Owns(x,z)]

- 3. No person would kill a cat.
  ¬[∃ x Person(x)∧ ∃y Cat(y) ∧ Kill(x,y)]

- 4. Some persons own a cat and a dog.
  ∃ x Person(x) ∧ ∃y Cat(y) ∧ Owns(x,y) ∧ ∃z Dog(z) ∧ Owns(x,z)

- 5. John owns only one bird.
 ∃ x Bird(x) ∧ Owns(John,x) ∧ ∀y [Bird(y) ∧ Owns(John, y) ⇒ x=y]

3   Question3

[10 points]

- 1. Write the knowledge given below in predicate calculus using the following predicates: Person(x) = x is a person; Rich(x) = x is rich; Happy(x) = x is happy; Read(x) = x can read; Exciting(x) = x has an exciting life; Stupid(x) = x is stupid The change made to Exciting is to make things easier. You are free to use the predicates I had before
  "All persons who are rich and are not stupid are happy. Persons who can read are not stupid. Happy persons have exciting lives. John is a person. John can read and he is rich. "

  - All persons who are rich and are not stupid are happy.
    ∀x[Person(x) ∧Rich(x) ∧¬Stupid(x)] ⇒Happy(x)

  - Persons who can read are not stupid.
    ∀x [Person(x) ∧ Read(x)]⇒¬Stupid (x)

- Happy persons have exciting lives.
  $\forall$x [Person(x) $\land$ Happy(x)] $\Rightarrow$Exciting(x)

- John is a person.
  Person(John)

- John can read and he is rich.
  Read(John) $\land$Rich(John)

- 2. convert to CNF
  1. $\neg$ Person(x) $\lor$ $\neg$Rich(x) $\lor$ Stupid(x) $\lor$ Happy(x)
  2. $\neg$ Person(x) $\lor$ $\neg$Read(x) $\lor\neg$Stupid (x)
  3. $\neg$Person(x) $\lor$ $\neg$Happy(x) $\lor$Exciting(x)
  4. Person(John)
  5. Read(John)
  6. Rich(John)

- 3. Answer the question "is there anyone who has an exciting life?" Notice that this is a question that requires not just True or False as an answer. Yes. Because John can read, and John is rich. so John is not stupid.Person who is rich and not stupid is happy and person who is happy has an exciting life.

4   Question4

[20 points] Prove by resolution that the following set of clauses (in predicate calculus) is unsatisfiable. Assume that upper case arguments are constant, lower case arguments are variable:

- 1. G(B)

- 2. $\neg$ G(x) $\lor$ H(x)

- 3. $\neg$ H(z) $\lor$ I(z)

- 4. $\neg$ H(w) $\lor$ J(w,D)

- 5. $\neg$ I(B) $\lor$ J(C,B)

- 6. $\neg$ I(q) $\lor$ $\neg$ J(q,y)
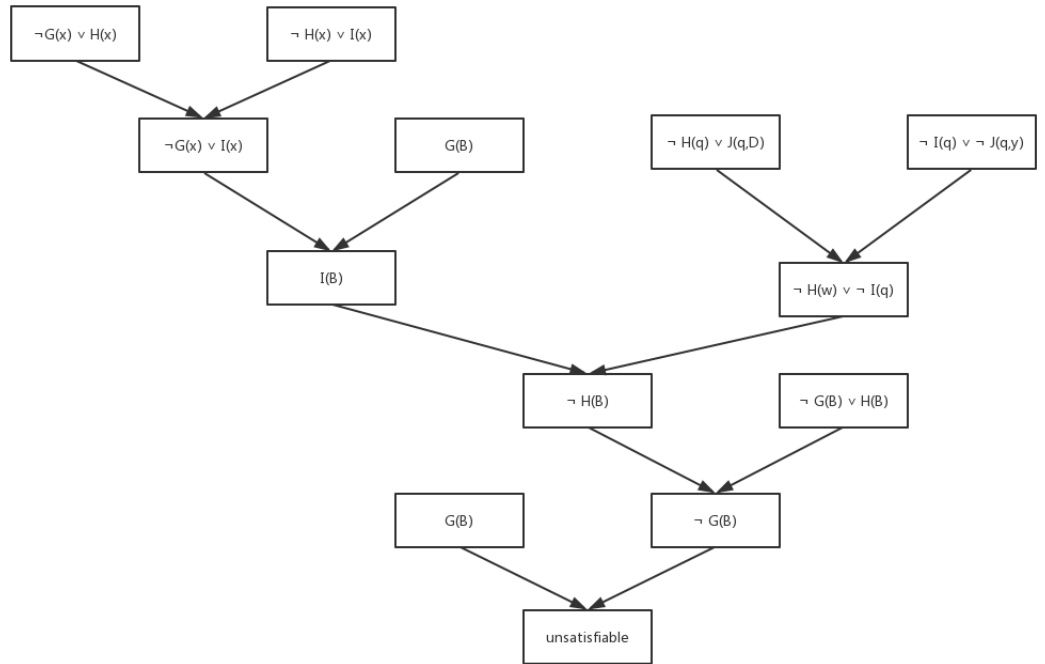
5 Question5

[30 points]

- 1. Write the following statements in predicate calculus:

    - A truck is a vehicle.
      $\lor$x Truck(x) $\Rightarrow$ Vehicle(x)

    - A SUV is a vehicle.
      $\lor$x SUV(x) $\Rightarrow$ Vehicle(x)

    - A car is a vehicle.
      $\lor$x Car(x) $\Rightarrow$ Vehicle(x)

    - A truck is bigger than a SUV.
      $\lor$x Truck(x) $\land$ $\lor$y SUV(y)$\Rightarrow$ Big(x , y)

    - There is a SUV that is bigger than every car.
      $\exists$x SUV(x) $\land$ $\lor$y Car(y) $\Rightarrow$ Big(x , y)

    - A RAM is a truck.
      Truck(RAM)

- A Tesla is a car.
  Car(Tesla)

- Bigger is transitive, i.e. if x is bigger than y and y is bigger than z
  then x is bigger than z. ∨x , y , z Big(x,y) ∧Big(y,z) ⇒Big(x,z)

- 2. Convert them to CNF. Pay attention to how you skolemize existentially
  quantified variables. Recall that a Skolem constant cannot be unified with
  another constant except itself, but it can be unified with a variable.

  - 1. ¬Truck(x) ∨ Vehicle(x)
  - 2. ¬SUV(x) ∨ Vehicle(x)
  - 3. ¬Car(x) ∨ Vehicle(x)
  - 4. ¬Truck(x) ∨¬SUV(y) ∨ Big(x , y)
  - 5. ¬SUV(x) ∨ ¬Car(y) ∨ Big(x , y)
  - 6. Truck(RAM)
  - 7. Car(Tesla)
  - 8. ¬Big(x,y) ∨¬Big(y,z) ∨Big(x,z)

- 3. Prove by resolution with refutation: "A RAM is bigger than a Tesla".
  Remember that a Skolem constant cannot be unified with another constant
  except itself, but it can be unified with a variable.

  - 1. ¬Truck(x) ∨ Vehicle(x)
  - 2. ¬SUV(x) ∨ Vehicle(x)
  - 3. ¬ Car(x) ∨ Vehicle(x)
  - 4. ¬Truck(x) ∨ ¬SUV(y) ∨ Big(x, y)
  - 5. Truck(RAM)
  - 6. Car(Tesla)
  - 7. ¬ Big(x,y) ∨ ¬ Big(y,z) ∨ Big(x, z)
  - 8. ¬Big(Truck(RAM), Car(Tesla))