Assignment #3
Artificial Intelligence - CSCE 523
Due: 8:00 AM, Monday February 24, 2020
Knowledge Representation and Planning

Turnin: E-mail me a zip file containing your typed solutions and associated files to all questions.

1. (5 points) What logic rule did the Cheshire Cat use in this argument, and is it sound?

   "To begin with," said the Cat, "a dog's not mad. You grant that?"
   "I suppose so," said Alice.
   "Well, then," the Cat went on, "you see a dog growls when it's angry, and wags its tail when it's pleased. Now I growl when I'm pleased, and wag my tail when I'm angry. Therefore I'm mad."

2. (5 points) Translate the following Lewis Carroll sentences into a Propositional Logic Knowledge Base and derive two statements from the Knowledge Base:

   All hummingbirds are richly colored.
   No large birds live on honey.
   Birds that do not live on honey are dull in color.

3. (5 points) Translate the following into First Order Logic and then convert to Conjunctive Normal Form (CNF):

   According to the Pidgeon: If little girls eat eggs, then they are a kind of serpent. Alice (who is a little girl) eats eggs. Therefore, she is a kind of serpent.

4. (10 points) Determine for the following pairs of sentences if they can be unified and if they can, given the most general unifier, if not discuss why.
   a. $P(x) \vee Q(Dog, x) \Rightarrow R(Cat,Dog)$
      $P(Cat) \vee Q(y,z) \Rightarrow R(z,y)$

   b. Queen(Hearts) ^ HasProblem(Hearts, y) ⇒ Solve(y, Beheading) ^ HasHeadandBody(y)
      Queen(x) ^ HasProblem(x, Cheshire) ⇒ Solve(y, Beheading) ^ HasHeadandBody(y)

   c. $((Son(x,x) \wedge Sister(Mary,Jack)) \Rightarrow (Daughter(x,Mary) \wedge Brother(Jack,Mary)$
      $((Son(Jack,x) \wedge Sister(z,x)) \Rightarrow (Daughter(z,f(x)) \wedge Brother(y,z)))$

5. (15 points) Use resolution with refutation to show that the following three queries can be inferred from the given knowledge base. At each resolution step also indicate the corresponding identifier and binding list.

   KB: S1: Uncle(John, Jack)
   S2: Aunt( Mary, Amy)
   S3: Female(Amy)
   S4: Brother(Jack, Amy)
   S5: Brother(Bill, Jack)
   S6: Sister(x,y) $\Rightarrow$ Siblings(x,y)
   S7: Brother(x,y) $\Rightarrow$ Siblings(x,y)
   S8: Brother(x,y) $\wedge$ Female(y) $\Rightarrow$ Sister(y,x)
   S9: Siblings(x,y) $\wedge$ Uncle(z,y) $\Rightarrow$ Uncle(z,x)
   S10: Siblings(x,y) $\Rightarrow$ Siblings(y,x)
   S11: Uncle(x,y) $\wedge$ Aunt(z,y) $\Rightarrow$ Married(x,z)
   S12: Uncle(x,y) $\wedge$ Married(z,x) $\Rightarrow$ Aunt(z,y)
   S13: Married(x,y) $\Rightarrow$ Married(y,x)

   a. Married(John, Mary)
   b. Aunt(Mary, Jack)
   c. $\exists$x(Siblings(Jack,x) $\wedge$ Uncle(John,x))

6. (10 points)Translate the knowledge base from problem 4 into a formula list for otter and use it to perform a proof by refutation for the queries from problem 4, and the two below. A copy of the otter executable and documentation can be found in the course directory. If you want to run otter on a non-Windows computer you can access the information you will need at http://www-unix.mcs.anl.gov/AR/otter/.
   If a sentence cannot be proven determine what needs to be added to the knowledge base to make it provable, and would this invalidate the KB. (Turn-in the otter files, and a copy of the screen output for each query)
   d. Brother(Bill, Amy)
   e. Uncle(John,Bill) ^ Siblings(Bill,Jack)

7. (25 points) For the following logic problem, a) encode the problem and have Otter solve it, and b) represent the problem as a constraint satisfaction problem and solve using the backward algorithm with forward checking.

   Link, Zelda, and Ganondorf fought three different evil creatures, the Octorock, an Iron Knuckle, and a Poe. They fought them with three different weapons, a bow and arrows, magic, and a sword. Determine who fought what creature and with what weapon.

1. Ganondorf did not fight the Octorock.
2. The Iron Knuckle was not fought against with magic.
3. Zelda fought the Octorock.
4. Link has a sword and did not fight the Poe.
5. Zelda fought with the bow and arrows.

8. (25 points) Use the Graphplan planning algorithm to solve the blocks world planning problem shown in Figure 1 and the Rush Hour problems in Figures 2 and 3. The executable, fact, and operator files for the blocks domain are in the course directory. You must modify the fact file to solve blocks world planning problem shown in Figure 1. And write your own fact file and operator file for the two Rush Hour problems shown in Figures 2 and 3. Assume there are no trucks only cars of length two. Define actions as a movement of a car one square north, south, east, or west depending on orientation. During execution of Graphplan, respond to the prompts to perform automatic time steps, and information, and for other hit 'B' for build until goals. Note your operator file (for Rush Hour) should be the same for both problems. The only part that should be different is the initial condition in the fact file.

   Turn in a copy of your domain files, and a printout of the solutions the planner found. For the Rush Hour problems also include a report on the search time required and a comparison on the ease of use between Graphplan, and the search you implemented in Assignment 1. Which makes the most sense for this domain? And why?

   Problem coding recommendations: The ops file will be fairly short; the work is in getting the fact file correct. For the facts create a car object for each car, and location for each grid square. Your preconditions should label whether vehicles can go horizontally or vertically, where the nose and tail of the car are located on the board, which squares are free, and what locations are horizontally and vertically adjacent to each other.

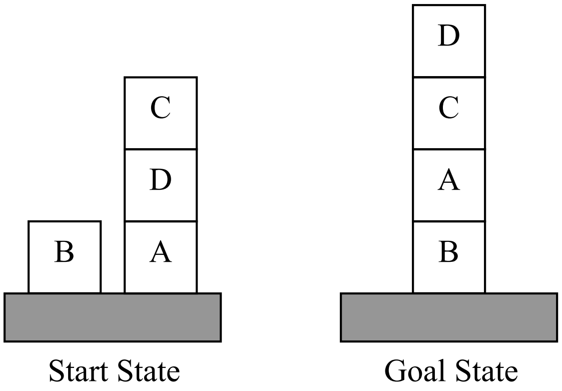

Figure 1: Blocks World Planning Problem.

```
....O1O1....
....A1......
X0X0A1......   <= EXIT
....B1B1....
............
............
```

Figure 2: Initial State Rush Hour Problem 1.

```
O1P1P1Q1Q1B1
O1....C1..B1
..X0X0C1....   <= EXIT
......D1....
......D1....
............
```

Figure 3: Initial State Rush Hour Problem 2.