CSC 520 - ARTIFICIAL INTELLIGENCE
ASSIGNMENT 3

Due Date : March 6th

This assignment consists of three questions which involve written answers and code. In order to complete the assignment you must submit a written report in pdf form detailing your answers to the questions as well as your code. As discussed in class all work must be your own. You may not use third party libraries or example code to complete the assignment.

Your answers to the conceptual questions must be uploaded to Moodle as a pdf file titled Assign3_UnityID.pdf. The PDF must be clear and well written. Your source code must be submitted as a self-contained zip called Assign3_UnityID.zip. All code must be clear, readable, and well-commented. Please include a README with instructions on how to run the code.

QUESTION 1 (20 Marks)

Consider the following English sentences.

1. Black tea is a type of tea and cheese is not.
2. Black tea can blend with Green Tea
3. If an element is a not a tea, then no tea can blend with it.
4. Some tea are unwilted and unoxidized.
5. Some tea can blend with all tea except the unoxidized ones.

a. (5 pts) Convert the sentences into first-order predicate logic. Use the following lexicon:

   tea (X) - X is a type of tea.
   unoxidize (X) - X has NOT been oxidized.
   wilt (X) - X has been wilted.
   blend (X, Y) - X can blend with Y.

b. (10 pts) Follow the steps in the textbook to convert the logic statements into CNF.
c. (5 pts) Using FOPL resolution, prove the conclusion that *"Black tea cannot blend with cheese"*.

Number your clauses, and indicate explicitly step-by-step what resolves together, and under what substitution.

QUESTION 2 (20 pts)

Use the lexicon:
s – A customer can shop at Costco
o – A Costco customer

i – A customer has a membership identification with him/her
d – Customers will get membership identifications
g – A customer has gold membership
si – A customer has silver membership
pre – A customer can enjoy premium discount
l – A customer can collect loyalty points
c – Carol is a customer
1. All Costco customers can collect loyalty points.
2. All Costco customers can get membership identifications.
3. Costco customers can shop at Costco stores only when they bring membership identifications with them.
4. Carol cannot shop at Costco stores.
5. Costco customers has either gold membership or Silver membership.
6. Carol is a Costco customer.
7. Carol has gold membership but doesn't get premium discount and cannot shop at Costco stores since she forgot to take her Costco identification.
8. Customers with gold memberships can have premium discount only if they have their identification with them while customers with silver memberships cannot.

a. (10 pts) Use propositional logic to determine if the specification is consistent. If the sentences are not consistent, use resolution to derive a contradiction. If they are consistent, use the truth-table method to show at least one model.
b. (5 pts) What if anything changes if Carol can shop at Costco stores and does the consistency change? Be specific in showing what's different.

QUESTION 3 (60 Marks)

This is a code question in which you are tasked with implementing MAC and Forward Checking to solve arbitrary versions of the N-queens problem. The N-queens problem is a classical constraint problem where the goal is to determine where queens can be placed on an $N * N$ chessboard such that no two queens threaten the other. A sample image for this problem is shown below:

0		Q			
1				Q	
2	Q				
3			Q		
4					Q
	0	1	2	3	4

Execution instructions:

For Python:

NQueens.py <ALG> <N> <ALG>_<N>

For Java:

javac NQueens.java

java NQueens <ALG> <N> <ALG>_<N>

where: ALG is one of either **FOR** or **MAC,** representing Backtracking search with **Forward Checking** or **Maintaining Arc Consistency** respectively. N represents the number of rows and columns in the chessboard as well as the number of queens to be assigned. <ALG>_<N> must be a text file to which you will write your solution.

When called your code must generate as many unique solutions for the problem as it can find up to up to 2*N using the specified algorithm. And as each solution is found it must write it to a text file. When the code completes it should report the number of solutions found, the real time taken, and the number of backtracking steps. As always your code should be clear, readable, and well-written.

The output in the file should be as follows:

The following solution is for N = 4

```
MAC/FORWARD CHECKING   ← Name of the Algorithm

Solutions : 2     ← Number of Solutions

Backtracks : 11   ← Number of Backtracks

#1
 0   0   1   0
 1   0   0   0
 0   0   0   1
 0   1   0   0
                  ← Solutions
#2
 0   1   0   0
 0   0   0   1
 1   0   0   0
 0   0   1   0
```

For implementation of the algorithms consider column 0 as the start position i.e place the queen in one of the possible positions in column 0 and propagate forward.