Assignment 3

Problem 1:

a. The constraint graph for this problem is shown in the file named problem1a.pdf. This constraint graph can, in fact, be used to simplify solving this problem, as it makes the connections between the different sections more clear. Basically, the constraint graph presents the problem in a somewhat clearer, easier to understand way, and can greatly speed up the process of searching for a solution.

b. For this problem I will refer to the top of the tree as the root, and I will refer the level below it level one, then level two, and so on. I am also assuming that the problem is only asking for the variable chosen at each level, but not what value that variable will be assigned. With that in mind, I have listed below the variable that is selected at each level, as well as why (whether it was selected due to MRV or the degree heuristic). Also, at each level of the tree, the next possible choices are going to only be sections that are connected to the section that is at the current level. If there are no more unchosen variables connected at the current level, the algorithm will choose from the remaining sections. The search will first use MRV to select a variable. If there exists a tie between two or more variables after using MRV, then the degree heuristic will instead be used. If another tie is reached, a variable will be randomly chosen.

Level 1: KA (degree = 6)
Level 2: TN (degree = 4)
Level 3: KE (MRV; possible values = 1)
Level 4: TL (Randomly chosen between M, TL, and AP because all three tied in the degree heuristic)
Level 5: AP (Randomly chosen over M; both had same degree and MRV)
Level 6: M (MRV; possible value = 1)
Level 7: G (MRV; possible value = 1)
Level 8: P (MRV; possible value = 2)
Level 9: SL (MRV; possible value = 3)

c. Here, KA is assigned the color red.

Steps for checking remaining legal values of remaining variables

1. Algorithm checks a neighbor of KA, and removes Red from their domain (G, M, TL, AP, TN, KE)
2. The arc connecting KA with the neighbor in step one is removed from the queue.
3. If a value was removed from the neighbor in step one, the arcs containing this neighbor are added to the queue, for checking later.
4. Go back to step one and repeat until no arcs remain in the queue. Eventually, there will be no more arcs containing KA.

d. The following is a list of color assignments that satisfies the problem:
KA — Red

TN — Green
KE — Blue
TL — Green
AP — Blue
M — Blue
G — Green
P — Blue
SL — Green

Problem 2:
function CHECK_EQUIVALENCE(KB1, KB2) returns an answer
answer <— true
for each sentence S1, S2 in KB1, KB2 (respectively) do
    if S1 NOT(=>) S2 OR S2 NOT(=>) S1
        answer <— false
return answer

Problem 3:
a. KB does not entail S1. There are combinations of A, B, and C where KB and S1 have two different truth values. Therefore, S1 cannot follow from KB, which means that KB does not entail S1.
b. NOT(KB) would simply be the opposite of KB. This means that each of it's possible truth values are switched. The same goes for NOT(S1). Taking the negative of both of them doesn't change the fact they will still both have different truth values for certain combinations of A, B, and C, and therefore NOT(KB) would not entail NOT(S1).

Problem 4:
(NOT(A) OR NOT(B) OR NOT(C) OR NOT(D)) AND (NOT(A) OR B OR NOT(C) OR D)

This CNF returns false for both the given cases.

Problem 5:
a. For the contract to be upheld, Mary must mow the lawn on May 3 only if it had rained on May 1 and if John gave her the $10,000 check on May 2.

Symbols:
R — Rain on May 1, 2017
C — $10,000 check given by John to Mary on May 2, 2017
M — Mary mows the lawn on May 3, 2017

The contract can be represented as follows:
(R => C) AND (C => M)

The contract doesn't say that John can't give a check if it didn't rain on May 1, and it also doesn't say that Mary can't mow the lawn if John didn't give her a check. The contract only states what each person MUST do if certain events happen. They are allowed to give a check/ mow the lawn even if it didn't rain/a check wasn't received.

b. The logical statement representing what actually happened:
NOT(R) AND C AND M

It did not rain on May 1, 2017, John gave Mary a $10,000 check on May 2, 2017, and then Mary mowed the lawn on May 3, 2017.

c. The contract was not violated. The contract implies that John is allowed to give Mary a $10,000 check on May 2 even if it didn't rain on May 1. Mary received the check, so by the contract, she must mow the lawn. She did mow the lawn, so her end of the contract was fulfilled, meaning that no violations of the contract occurred.

Problem 6:
A Horn clause is one that contains one or zero positive symbols.
The given KB is not in Horn Form, as there are clauses with more than one positive literal. It also cannot be converted into Horn Form as A and E (at the end of the KB) are not in conjunction with any other symbols, and they are positive. This means they can't be transformed into proper Horn clauses, making a full conversion to Horn Form impossible.

In the statement D, only D is true, and all other variables are false. In order for a statement to follow from the KB, A and E must be true. Therefore, the KB does not entail the statement D.

Backwards chaining isn't really possible as the only clause that contains D is (B AND C) => D. That means that, even if D is true, the clause (B AND C) may be false. And plus, the fact that a relation between D and A or E cannot be shown also further implies that backwards chaining is impossible for the statement D. Either way, as shown above, KB does not entail D.

Problem 7:
a. (A <=> (B AND C))
(A => (B AND C)) AND ((B AND C) => A)
(NOT(A) OR (B AND C)) AND (NOT(B AND C) OR A)
(NOT(A) OR (B AND C)) AND ((NOT(B) OR NOT(C)) OR A)
(NOT(A) OR B) AND (NOT(A) OR C) AND ((A OR NOT(B)) OR (A OR NOT(C)))

b. (NOT(B AND A)) => C
NOT(NOT(B AND A)) OR C
NOT(NOT(B) OR NOT(A)) OR C
(A AND B) OR C