Theory Questions

1.Which of the following are true and which are false? Give brief explanations.

a. In a fully observable, turn-taking, zero-sum game between two perfectly rational players,it does not help the first player to know what strategy the second player is using—that is, what move the second player will make, given the first player's move.

b. In a partially observable, turn-taking, zero-sum game between two perfectly rational players, it does not help the first player to know what move the second player will make, given the first player's move.

c. A perfectly rational backgammon agent never loses.

See textbook Chapter6 24.6

your answer here...

a:True. The second player will play optimally, and so is perfectly predictable up to ties. Knowing which of two equally good moves the opponent will make does not change the value of the game to the first player.

b:False. In a partially observable game, knowing the second player's move tells the first player additional information about the game state that would otherwise be available only to the second player.

c:False. Backgammon is a game of chance, and the opponent may consistently roll much better dice. The correct statement is that the expected winnings are optimal.

2.Define in your own words the terms constraint, backtracking search, arc consistency,backjumping, min-conflicts, and cycle cutset.

See textbook Chapter6 6.16

your answer here...

constraint： A constraint is a restriction on the possible values of two or more variables.

backtracking search: Backtracking search is a kind of depth-first search (DFS). For a certain search tree (search tree is the role of recording path and state judgment), backtracking and DFS, the main difference is that backtracking method is not in the solution process. The complete tree structure is preserved, while the depth-first search records the complete search tree.

arc consistency:A directed arc from variable A to variable B in a CSP is arc consistent if, for every value in the current domain of A, there is some consistent value of B.

backjumping:Backjumping is a way of making backtracking search more efficient, by jumping back more than one level when a dead end is reached.

Min-conflicts is a heuristic for use with local search on CSP problems. The heuristic says that, when given a variable to modify, choose the value that conflicts with the fewest number of other variables.

cycle cutset: A cycle cutset is a set of variables which when removed from the constraint graph make it acyclic (i.e., a tree). When the variables of a cycle cutset are instantiated the remainder of the CSP can be solved in linear time.

3.Explain why it is a good heuristic to choose the variable that is most constrained but the value that is least constraining in a CSP search.

See textbook Chapter6 6.9

your answer here...

The most constrained variable makes sense because it chooses a variable that is (all other things being equal) likely to cause a failure, and it is more efficient to fail as early as possible (thereby pruning large parts of the search space). The least constraining value heuristic makes sense because it allows the most chances for future assignments to avoid conflict.
4.Consider the following procedure for choosing moves in games with chance nodes:
1、Generate some dice-roll sequences (say, 50) down to a suitable depth (say, 8).
2、With known dice rolls, the game tree becomes deterministic. For each dice-roll sequence,solve the resulting deterministic game tree using alpha-beta.
3、Use the results to estimate the value of each move and to choose the best.Will this procedure work well? Why (or why not)?
See textbook Chapter5 5.19

your answer here...

This procedure will give incorrect results. Mathematically, the procedure amounts to assuming that averaging commutes with min and max, which it does not. Intuitively, the choices made by each player in the deterministic trees are based on full knowledge of future dice rolls, and bear no necessary relationship to the moves made without such knowledge.
5. Now,please consider this game: there are three plates A, B and C, each plate has three bills. A puts 1, 20, 50; B puts 5, 10, 100; C puts 1, 5, 20. All units are "Yuan". There are two persons A and B, and two of them can check out three plates and banknotes .（A is ourself, The other is B） The game is divided into three steps:

1、A select a plate from three plate.
2、B take out two banknotes from A selected plate,and give the banknotes to A.
3、A take one of the two banknotes, and take it away.among,A want to get the max banknotes, B want to let A to get the min.
Try to understand the minimax algorithm , you can click the link:https://blog.csdn.net/tangchenyi/article/details/22920031

your answer here...
1、The following figure is the pattern tree of the above example problem:

2、then our choose the banknotes, so we should select the maximum value structure, the value of each node is the maximum .

3、The third step, assume that the other side will try to give us the minimized value.We can also

choose the corresponding maximum

4、The value of the root node is 20, which means that the opponent makes perfectly of every decision,This is our best decision under the Minimax algorithm. The pattern transformation path is shown in the red path :

Minimax is generally looking for a local optimal solution rather than a global optimal solution. The more the search depth is more likely to find a better solution, but the time consuming will be exponentially expanded.