CPSC 322 Midterm Practice Questions

The midterm will contain two long answer questions (sample in another file). Otherwise, it will be made up of approximately 6 questions like the ones in the list below. Some of the 6 questions may actually come verbatim from the list below. Unless otherwise noted, answers to the following questions should be a few sentences long.

1 What is AI?

1. **(What is AI?)**

   Compare and contrast an AI system that thinks and acts humanly with one that thinks and acts rationally?

2. **(What is AI?)** Describe the difference between a deterministic representation and a stochastic representation for an AI system.

3. **(What is AI?)** What is the difference between a goal and a preference function? Provide an example each.

2 Search

1. **(Search)** State the five elements that characterize a search algorithm.

2. **(Search)** Explain what the frontier is in a graph-search problem, and what it is used for.

3. **(Search)** Where b is the maximum branching factor and d is the maximum depth of the search, characterize the maximum size of a DFS frontier.

4. **(Search)** Is the worst-case time complexity different for DFS and BFS? Why or why not?

5. **(Search)** Consider the following generic search algorithm:

```
1   Input: a graph,
2          a set of start nodes,
3          Boolean procedure goal(n) that tests if n is a goal node.
4   frontier := {⟨s⟩ : s is a start node};
5   while frontier is not empty:
6          select and remove path ⟨n_0, ..., n_k⟩ from frontier;
7          if goal(n_k)
8            return ⟨n_0, ..., n_k⟩;
9          for every neighbor n of n_k
10           add ⟨n_0, ..., n_k, n⟩ to frontier;
11  end while
```

   Imagine that this generic algorithm was used as the basis for an implementation of depth-first search and of breadth-first search. Which line or lines of the pseudocode above *must* have different implementations? Briefly, how would those implementations differ?

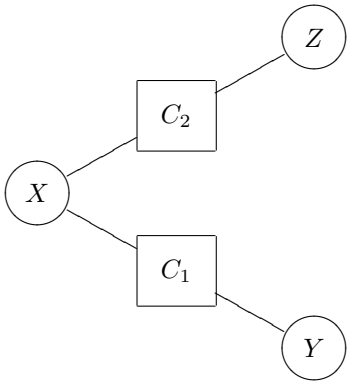

Figure 1: An abstract constraint network

6. **(Search)** Give the definition of an admissible heuristic.
7. **(Search)** In what sense is Breadth First better than Depth First? In what sense is Depth First better than Breadth First?
8. **(Search)** In what sense is branch and bound better than A*? In what sense is A* better than branch and bound?
9. **(Search)** Is the sum of two admissible heuristics also admissible? Why or why not?
10. **(Search)** Is the max of two admissible heuristics also admissible? Why or why not?
11. **(Search)** Is the min of two admissible heuristics also admissible? Why or why not?
12. **(Search)** Explain the distinction between optimality and optimal efficiency for search algorithms.
13. **(Search)** Assume you run uninformed iterative deepening to find solution no more than k steps from the start node ($k > 2$). In the worst case, how many times the nodes two steps from the start node will be generated? why?
14. **(Search)** Assume uninformed iterative deepening is running. If the algorithm has reached the stage in which DFS is running with bound depth 6: must it be true that the start state is not the goal? Why? must it be true that there are no solution at level 3? Why?

3 CSPs

1. **(CSPs)** How many states can be represented by eight variables, each of which can take four values?
2. **(CSPs)** How many states can be represented by three variables, each of which can take three values? How many states are there for a 3-queen problem?
3. **(CSPs)** State the condition under which an arc $\langle X, r(X, \bar{Y})\rangle$ is arc consistent.
4. **(CSPs)** Consider the constraint graph in Figure 1. Imagine that arc consistency has just reduced the domain of X as a result of considering the edge $\langle X, C_1\rangle$. Do we have to add the edge $\langle X, C_2\rangle$ back to the list of "to-do arcs"? Why or why not?
5. **(CSPs)** Consider the constraint graph in Figure 1. Imagine that arc consistency has just reduced the domain of X as a result of considering the edge $\langle X, C_1\rangle$. Do we have to add the edge $\langle Y, C_1\rangle$ back to the list of "to-do arcs"? Why or why not?
6. **(CSPs)** Consider the constraint graph in Figure 1. Imagine that arc consistency has just reduced the domain of X as a result of considering the edge $\langle X, C_1\rangle$. Do we have to add the edge $\langle Z, C_2\rangle$ back to the list of "to-do arcs"? Why or why not?

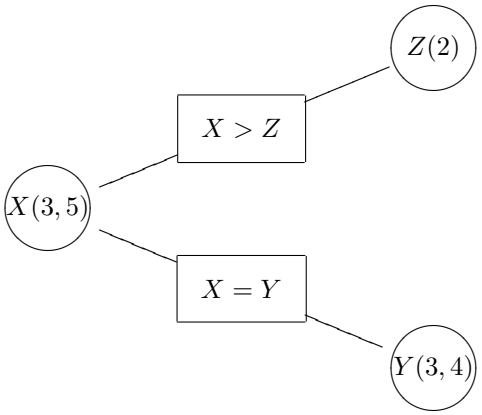

Figure 2: A concrete constraint network

7. **(CSPs)** How can a CSP be solved by search (specify states , neighbors, start state, goal state). What search strategy would you use? Why?

4 Local Search

1. **(Local Search)** What is the state space of local search for a CSP?
2. **(Local Search)** What are two key limitations of local search?
3. **(Local Search)** Explain a reasonable way to generate neighbors for local search ?
4. **(Local Search)** Define a plateau. Why are plateaus a problem for local search?
5. **(Local Search)** Why does randomization help in local search?
6. **(Local Search)** Explain how the next state is chosen in simulated annealing.
7. **(Local Search)** How would you change a local search algorithm to better find global maxima that are surrounded by local maxima? Assume that the neighbour relation is fixed.
8. **(Local Search)** How would you design a local search algorithm for a problem where there are no plateaus and no local minima that are not global minima? Assume that the neighbour relation is fixed.
9. **(Local Search)** Give an example of a crossover algorithm that could be used in a genetic algorithm; do not use the example given in class.
10. **(Local Search)** What is the difference between a random step and a random restart?
11. **(Local Search)** Consider two local search algorithms, A and B. A solves $x\%$ of the problems it is given within x^2 minutes. B solves $y\%$ of the problems it is given within y minutes. Is one algorithm always better than the other? If not, which algorithm would you use when?
12. **(Local Search)** Consider two local search algorithms, A and B. A solves $\sqrt{x}\%$ of the problems it is given within x minutes. B solves $y\%$ of the problems it is given within y minutes. Is one algorithm always better than the other? If not, which algorithm would you use when?

5 Planning

1. **(Planning)** What is the STRIPS assumption?
2. **(Planning)** In forward planning what is the branching factor for a given state?

3. **(Planning)** What are the two parts of an action in STRIPS, and what is the meaning of each part?

4. **(Planning)** What is a node in the search space of forward planning?

5. **(Planning)** Explain the role of the horizon in CSP planning. How does the choice of horizon affect the behavior of the planner?

6. **(Planning)** In a CSP encoding of a planning problem, what is the relationship between the number of variables in the CSP (n), the number of variables (v) and actions (a) in the planning problem, and the horizon h?

7. **(Planning)** In a CSP encoding of a planning problem, which variables are involved in an effect constraint?