CPSC 322 **Artificial Intelligence I** **Fall 2020**

Sample Midterm Exam #1

The exam will be worth 60 marks and take 75 minutes. This means that we expect you should spend one minute per mark. Each question below has an associated mark, so you can calibrate your timing. (So this practice midterm should take you longer than the 75 minutes of the exam). Short answer questions require at most two sentences to explain each point, unless they specify otherwise. Always write full sentences if you are asked for an explanation.

During the exam, you may use programs and the Internet, but you are not allowed to consult or talk to anyone. You will not have time during the exam to represent all of the problems in AIPython or AISpace; instead you need to apply your understanding to a particular problem. We reserve the right to give students an oral exam to explain their working — this is both to detect cheating and to help you. If you did the work yourself, the oral exam will not harm your grade.

The midterm will cover (some of):

- AI and agents, including what is AI, definition of an agent, and the agent design dimensions.
- States and searching.
- CSPs: search, arc consistency with domain splitting, local search.

It will only cover sections of the textbook that are listed under the schedule tab of the course web page. It will not include planning.

You can expect to have questions that arise from the assignments, so look at all ow the questions and solutions (even those you did not do).

If the question asks for a definition, give the definition in your own words. Don't plagiarize from the textbook or any other source. (We are not testing your ability to copy from the Internet.) Copying verbatim is plagiarism and will not be tolerated. If asked to explain something in the context of a particular case, giving a generic definition will not be worth any marks.

1 Agents

(a) (i) [3 marks] As an input to the agent, what is "abilities"?
(ii) [5 marks] Suppose Pat claimed that the abilities should not be one of the inputs to the agent, as it does not affect the actions. Give an example showing why Pat is wrong.

(b) [5 marks] Describe the difference between a fully-observable deterministic system and a partially-observable stochastic system.

(c) [5 marks] What is the "planning horizon"? Describe the difference between a finite stage and indefinite stage planning horizon.

(d) [5 marks] Give a specific examples of an ordinal preference. How is an ordinal preference different from a goal?

2 Search

(a) [5 marks] Explain what the frontier is in a graph-search problem, and what it is used for.

(b) [3 marks] Is the worst-case time complexity different for depth-first search and breadth-first search? (You must be specific about what the complexity is with respect to.) Why or why not?

(c) [5 marks] 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?

(d) [4 marks] In what sense is branch and bound better than A*? In what sense is A* better than branch and bound?

(e) [3 marks] Is the maximum of two admissible heuristics also admissible? Why or why not?
Is the sum of two admissible heuristics also admissible? Why or why not?

(f) [10 marks] Consider the cyclic delivery graph of Figure 3.7 of the textbook, with the heuristic function of Example 3.13. [Of course, in the exam, we will give you the graph and the heuristic costs.] Suppose the start node is $b1$ and the goal is $r123$.

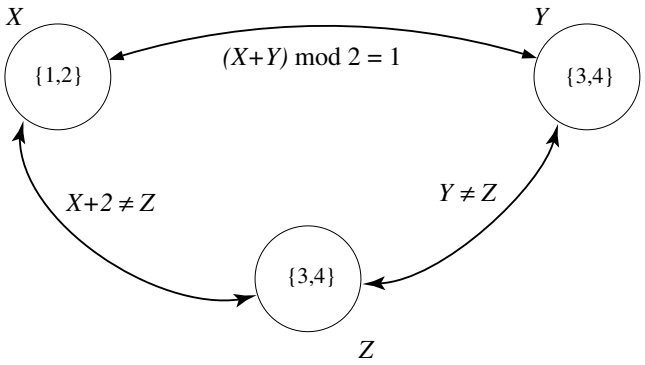
For each iteration of A^*, a trace would show the path removed from the frontier and the path(s) added to the frontier to get from $b1$ to $r123$. We have started the trace, please continue it. You only need to show **three iterations**; that is, for the next three values selected from the frontier show which paths are added to the frontier (together with their *cost*-value, h-value and f-value). [The path number has no meaning; it is just a number so you can refer to it.]

Path Number	Path	*cost*	h	f
1.	$b1$	0	18	18
Remove path number 1.				
2.	$b1 \rightarrow b2$	6	15	21
3.	$b1 \rightarrow b3$	4	17	21
4.	$b1 \rightarrow c2$	3	10	13

(g) [3 marks] For this particular example, when is the first time that multiple-path pruning will prune a path? Be specific about which path is pruned and why.

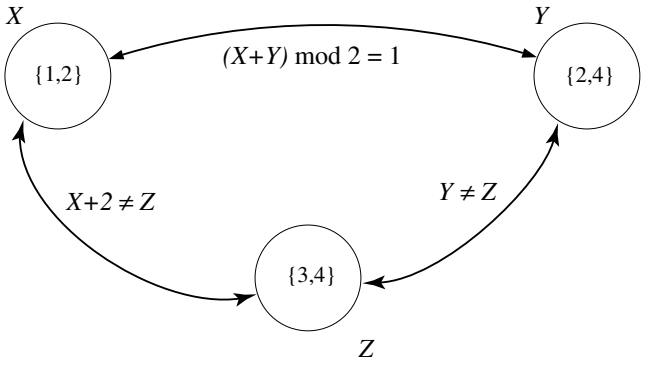
3 Constraints

(a) [5 marks] Consider the following constraint graph. Note that $(X + Y) \bmod 2 = 1$ means that $X + Y$ is odd.



Is it arc consistent? If it is, explain why. If it isn't, which domain element(s) that can be pruned? For each domain element(s) that can be pruned, give the arc (or arcs) that can be used to prune it.

(b) [5 marks] Consider the following constraint graph:



Is it arc consistent? If it is, explain why. If it isn't, which domain element(s) that can be pruned? For each domain element(s) that can be pruned, give the arc (or arcs) that can be used to prune it.

(c) [3 marks] Why does randomization help in local search?

(d) [3 marks] Explain how the next assignment is selected in simulated annealing.

(e) [5 marks] What is a runtime distribution? What is on the x-axis? What is on the y-axis? How is a runtime distribution constructed?

(f) [3 marks] How can a runtime distribution be used to show that one local search algorithm dominates another? If one algorithm dominates another is it always better?

(g) [2 marks] Given a runtime distribution, how can the median runtime be determined?

(h) [3 marks] Why is the mean runtime not a good measure for the performance of a local search algorithm?