CMPT 310 - Artificial Intelligence Survey

Assignment 2

Due date: March 1, 2021 — J. Classen
10 marks — February 9, 2021

Important Note: Students must work individually on this, and other CMPT 310, assignments. You may not discuss the specific questions in this assignment, nor their solutions with any other student. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the general concepts involved in the questions in the context of completely different problems. If you are in doubt as to what constitutes acceptable discussion, please ask!

Question 1 (2 marks)

For each of the following assertions, say whether it is true or false and give a brief justification (i.e., no more than 1–3 sentences) to support your answer.

(a) An agent that senses only partial information about the state cannot be perfectly rational.

(b) There exist task environments in which no pure reflex agent can behave rationally.

(c) There exists a task environment in which every agent is rational.

(d) Every agent function is implementable by some program/machine combination.

(e) Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.

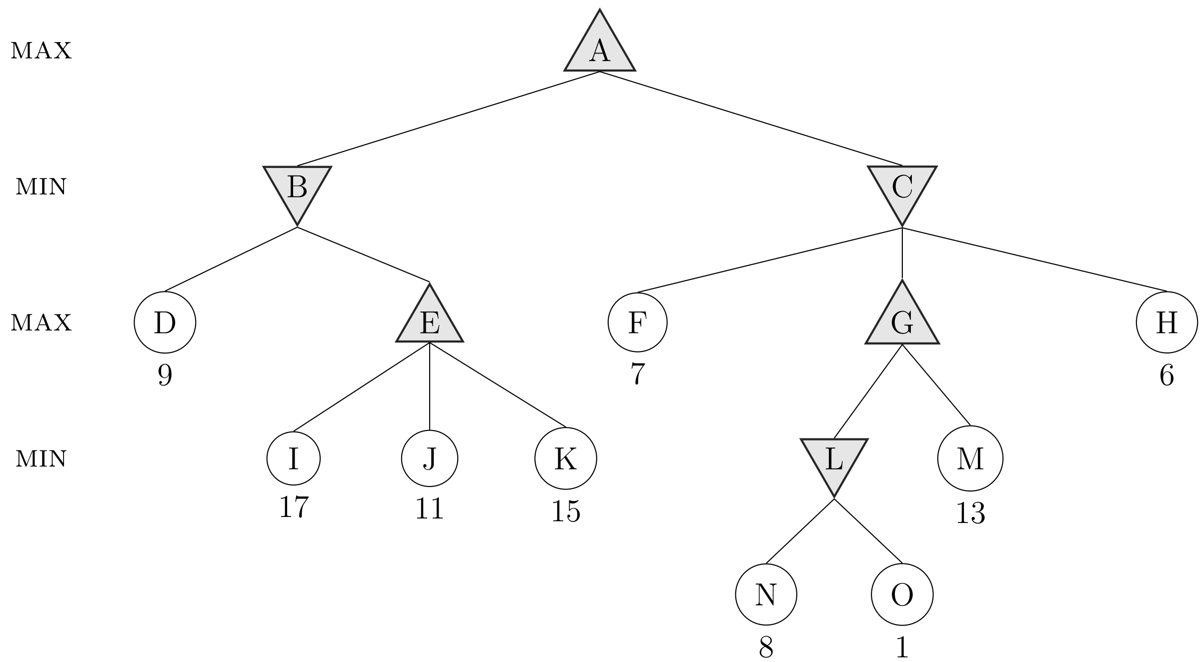
(f) It is possible for a given agent to be perfectly rational in two distinct task environments.

(g) Every agent is rational in an unobservable environment.

(h) A perfectly rational poker-playing agent never loses.

Question 2 **(3 marks)**

Consider the MAX-MIN game tree shown below where the numbers underneath the leaves of the tree are utility values from the first player's point of view (MAX).



(a) Draw a copy of the tree on paper and perform the **minimax** algorithm on it by hand. Write the resulting minimax values next to every node.

(b) Do the same, but with **left-to-right alpha-beta** pruning. Write the final values for α and β next to every node, and indicate which nodes are not examined due to pruning.

(c) Do the same, but with **right-to-left alpha-beta** pruning. Again, show the final values for α and β, and indicate which nodes are not examined.

Question 3 (3 marks)

Consider the following two-player game (players A and B):

A			B
1	2	3	4

The starting position of the simple game is shown on the right: Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1 or forward to 4.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$; if player B reaches space 1 first, then the value of the game to A is -1.

(a) Draw the complete game tree using the following conventions:

- Annotate each terminal state with its game value in a circle.
- Treat loop states as terminal states. Since it is not clear how to assign values to loop states, annotate each with a question mark in a circle.

  Loop states are states that already appear on their path to the root at a level in which it is the same player's turn to move.

(b) Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the question mark values and why.

(c) Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to part (b). Does your modified algorithm give optimal decisions for all games with loops?

(d) Which player has a winning strategy, and what does it look like? And what can be said about the general case, when instead of a 4-square game, we consider an n-square game for $n > 2$?

Question 4 **(2 marks)**

Consider a vocabulary with the following symbols:

- $Occupation(x, y)$: Predicate. Person x has occupation y.
- $Customer(x, y)$: Predicate. Person x is a customer of person y.
- $Boss(x, y)$: Predicate. Person x is a boss of person y.
- $Doctor$, $Surgeon$, $Lawyer$, $Actor$: Constants denoting occupations.
- $Emily$, Joe: Constants denoting people.

Use these symbols to write the following assertions in first-order logic:

(a) Emily is a surgeon or a lawyer.

(b) Joe is an actor, but he also holds another job.

(c) All surgeons are doctors.

(d) Joe does not have a lawyer (i.e., is not a customer of any lawyer).

(e) Emily has a boss who is a lawyer.

(f) There exists a lawyer all of whose customers are doctors.

(g) Every surgeon has a lawyer.