**MSBD5015 Artificial Intelligence**
**Fall 2022 Final**
**1500 on 10/12/2022**
**Time Limit: 3 hours**

**Name:** ______________________

**Stu ID:** ______________________

**Instructions**:

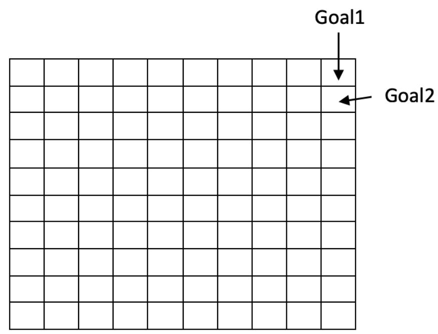
1. This exam contains 14 pages (including this cover page) and 12 questions.
2. This is a closed book exam.
3. Observe the honor code. Write the exam on your own.

Grade Table (for teacher use only)

| Question | Points | Score |
|---|---|---|
| Reactive Agents and State Machines | 10 | |
| Genetic Programming | 5 | |
| Search Problem Formulation | 5 | |
| MDP | 5 | |
| Heuristic Search | 10 | |
| Alpha-Beta Pruning | 5 | |
| FOL Representation | 10 | |
| Propositional Logic | 10 | |
| Game theory and auction | 10 | |
| Uncertainty | 10 | |
| Perceptron and GSCA Learning | 18 | |
| Reinforcement Learning by ChatGPT | 2 | |
| Total: | 100 | |

**Question 1: Reactive Agents and State Machines .................... 10 points**

Consider the robot with the same specification as our boundary-following robot discussed in class: eight sensors $s_1, ..., s_8$ and four actions (*North*, *South*, *East*, and *West*). Now suppose the environment is a 10x10 grid without any obstacles as shown below:




For each of the two goals indicated in the figure, can it be achieved by a reactive agent? By achieving a goal we mean that whatever the robot's initial position, the robot will get to the goal position, and then stop (*nil* action). For each of the two goals:

- If your answer is yes, give a production system for it. Do not call another production system. Write your own rules.
- If your answer is no, give your reason for it, and state how many steps the robot needs to remember in order to achieve the goal. No need to give a formal proof. An informal short explanation will be suffice. By k steps memory, we mean the robot remembers her past k actions and sensor readings. So 1 step memory means remembering last action and what the last sensor readings were.

**Question 2: Genetic Programming** .................................... **5 points**

Consider using genetic programming to evolve a program to control the robot to follow the boundary in the environment given in the last question. To do that, we need a fitness function to compute the fitness of any given program. Which of the following ones is most suitable, and why?

1. Ramdomly choose 10 starting positions, and for each of them, execute the given program until it has carried out **20** steps. The fitness value of the program is then the total number of next-to-wall cells visited during these executions.
2. Ramdomly choose 10 starting positions, and for each of them, execute the given program until it has carried out **200** steps. The fitness value of the program is then the total number of next-to-wall cells visited during these executions.
3. Ramdomly choose 10 starting positions, and for each of them, execute the given program until it has carried out **2,000,000** steps. The fitness value of the program is then the total number of next-to-wall cells visited during these executions.

**Question 3: Search Problem Formulation** ............................... **5 points**

Consider the missionaries and cannibals problem:

> In the missionaries and cannibals problem, three missionaries and three cannibals must cross a river using a boat which can carry at most two people, under the constraint that, for both banks, if there are missionaries present on the bank, they cannot be outnumbered by cannibals (if they were, the cannibals would eat the missionaries). The boat cannot cross the river by itself with no people on board.

Assume that initially the missionaries and cannibals are at the left bank, and the goal is to get them all to the right bank safely. Also assume that the boat is initially at the left bank, and we do not care where it is in the goal state. Formulate this problem as a search problem by defining: states, initial state, goal condition (for checking if a state is a goal), operators and their costs.

**Question 4: MDP** .................................................... **5 points**

Consider again the missionaries and cannibals problem in the last question. Now there are two uncertainties:

- Even if the number of cannibals ($C$) is greater than the number of missionaries ($M$), it may still be safe: if $C > M$, then with probability $M/C$ it is safe and probability $(C - M)/C$ it is unsafe (i.e. the missionaries got eaten).
- The boat is 100% safe for one person but has a 10% probability of sinking if two people are in it.

Formulate this setting as an MDP by giving the following:

- states;
- starting state;
- terminating condition $End(s)$;
- reward function $R(s, a, s')$: -1 if the boat sinks or the state is unsafe, 1 if all of them have crossed the river (i.e. all of them are on the right bank), 0 otherwise (meaning we don't care how long it takes for them to cross the river).
- discount fact: Let it be 1.
- state transition relation $T(s, a, s')$: for the sake of time, define this relation only for the starting state $s$.

**Question 5: Heuristic Search ........................................ 10 points**

Consider the following state space. The number next to a state is the value of the heuristic function on the state, and the number next to an arc is the cost of the corresponding operator. If the number next to a state is 0, that means that the state is a goal state.

(5.1) Give the sequence of the states expanded by $A^*$ algorithm, starting from the root $A$ and terminating at a goal state. Notice that whenever there is a tie, we prefer nodes at deeper levels, and on the same level, left to right.

(5.2) Can you adjust the heuristic function so that the goal state $G$ is returned? Notice that your heuristic function still needs to be admissible, and you don't need to do this question if $G$ is already the terminating goal state in your solution to (5.1).

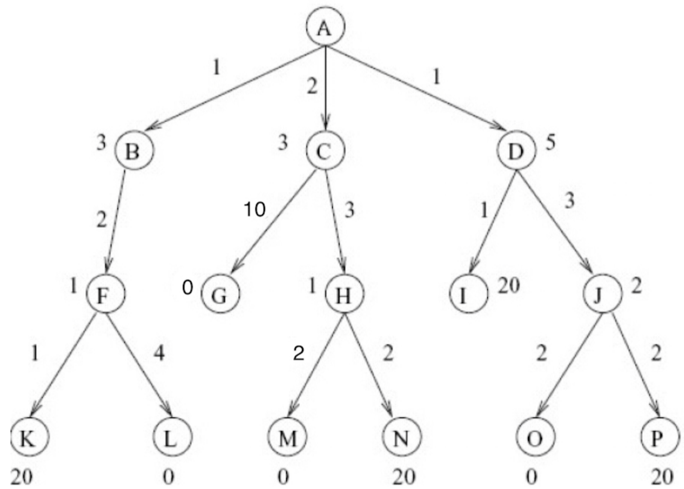



Figure 1: A search problem

**Question 6: Alpha-Beta Pruning** .................................... **5 points**

Perform a left-to-right alpha-beta prune on the following minimax game tree. Which nodes are pruned? What's the value of the root? Left-to-right means that whenever a node is expanded, it's children are considered in the order from left to right. This means the leaf nodes are generated from left right. So the first leaf node considered is D, followed by E, followed by M and so on.

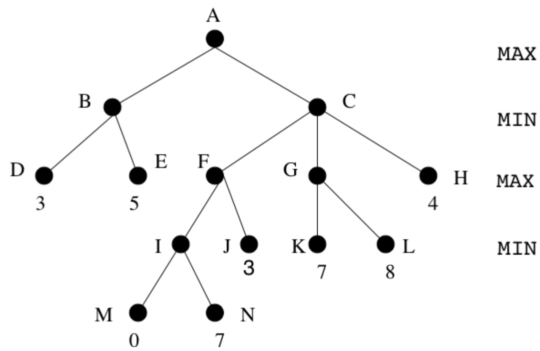



Figure 2: A minimax search tree

**Question 7: FOL Representation .................................... 10 points**

Let $Fools(x, y, t)$ stands for "$x$ can fool $y$ at time $t$". Represent the following statements in first-order logic:

1. Everyone can fool every other person sometimes. Notice that this means that for example, Jane can fool both Jack and John but not necessarily at the same time.
2. There is exactly one time when someone can fool everyone else.
3. At no time can someone fool all other people.
4. A person cannot fool another person all the time.
5. For any two people $x$ and $y$, if $x$ can fool $y$ some time, then $y$ can also fool $x$ some time.

**Question 8: Propositional Logic** .......................................... **10 points**

A, B, and C are three friends, and we know:

1. A drinks tea.
2. B drinks wine and dates C.
3. C drinks either tea or wine but not both, and dates A.

Assuming the following propositions:

- $T_x$: $x$ drinks tea. So this yields three propositions: $T_A$, $T_B$, and $T_C$.
- $W_x$: $x$ drinks wine. This will yield three propositions.
- $D_{x,y}$: $x$ dates $y$. This will yield 9 propositions.

Answer the following questions:

(8.1) Represent the above facts as propositional formula using the given propositions.

(8.2) Convert your formulas to clauses.

(8.3 ) Represent the following query as a formula: A wine drinker dates a tea drinker. To make your formula shorter, you can assume that by a wine drinker, we mean either B or C, and by a tea drinker either A or C. Also no one dates themselves.

(8.4) Convert the negation of your query formula to clauses.

(8.5) Use resolution to derive the empty clause from all the clauses (those from your facts and from the negation of your query).

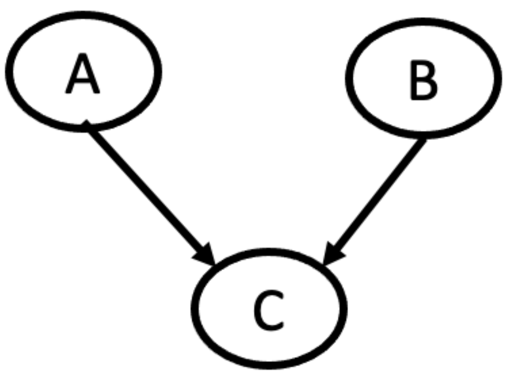
**Question 9: Game theory and auction ................................ 10 points**

Consider the single item first-price auction with two bidders $B_1$ and $B_2$, and assume that when there is a tie, $B_1$ will win. Suppose $B_i$ values the item $x_i$, $i = 1, 2$, and the information is common knowledge (thus $B_1$ knows that $B_2$'s value is $x_2$, and vice versa and so on). Suppose further that each can bid with any integer in the interval $[1, 100]$.

- Make this auction into a game in normal form $(\{B_1, B_2\}, R_1, R_2, u_1, u_2)$ by defining $R_i$ (the set of actions for player $B_i$) and $u_i$ (player $B_i$'s utility function). You can assume that both players are risk neutral.
- Suppose $x_1 = 10$ and $x_2 = 5$. What are the Nash equilibria of your game?

**Question 10: Uncertainty .............................................. 10 points**

Consider the following Bayesian network:




Prove the two semantics for this network is equivalent. That is, prove that the following two are equivalent:

1. $P(A, B, C) = P(A)P(B)P(C|A, B)$.
2. $A$ and $B$ are independent.

**Question 11: Perceptron and GSCA Learning ........................ 18 points**

Consider the following data set:

| ID | $x_1$ | $x_2$ | $x_3$ | $x_4$ | OK |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | Yes |
| 2 | 0 | 1 | 1 | 0 | Yes |
| 3 | 0 | 0 | 1 | 0 | No |
| 4 | 1 | 1 | 0 | 1 | No |

where $x_1$, $x_2$, $x_3$, and $x_4$ are some features that should not concern us here.

(11.1) Run the perceptron error-correction procedure on this dataset using the learning rate = 1, and the initial weights all equal to 0. Recall that the threshold is considered to be a new input that always have value "1". Please give your answer by filling in the following table, where weight vector $(w_1, w_2, w_3, w_4, t)$ means that $w_i$ is the weight of input $x_i$, and $t$ is the weight for the new input corresponding to the threshold. Run it for 3 iterations.

| ID | Weight vector | Weighted Sum | Actual | Desired |
|---|---|---|---|---|
| Initial | 0, 0, 0, 0, 0 | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

(11.2) Does it converge? If so, give the Boolean function corresponding to the perceptron. If not, either give a perceptron that agrees with the training set thus showing that the procedure will converge but needs more iterations, or prove that there is no perceptron that can agree with the training set, i.e. the training set is not linearly separable, so the error-correction procedure will not converge. [HINT To show that the training set is not linearly separable, prove by refutation: suppose there is one, then the four examples in the training set will yield four contraints that are inconsistent, like how we proved the exclusive or is not linearly separable in class.]

(11.3) From the same training set, apply the GSCA algorithm to try to learn a set of rules. Give the set of rules if it succeeds. If it fails to learn a set of rules, explain why it failed.

**Question 12: Reinforcement Learning by ChatGPT .................. 2 points**

I asked ChatGPT, a recently released natural langauge conversational agent by OpenAI, to give me "a good exam question for reinforcement learning." Below is what ChatGPT returned:

> Consider a reinforcement learning agent interacting with an environment. The agent has a set of possible actions, $A$, and the environment has a set of possible states, $S$. At each time step $t$, the agent receives a reward $R_t$ from the environment and observes the current state $s_t$. The goal of the agent is to learn a policy, $\pi$, that maps states to actions, such that the expected cumulative reward over time is maximized.

Suppose the agent is currently in state $s$ and is considering two possible actions: $a_1$ and $a_2$. The agent has observed the following rewards when taking these actions in state $s$:

- $a_1$: 3, 2, 5, 4, 7
- $a_2$: 4, 1, 1, 6, 1

Which of the following statements is true?

(a) The agent should choose $a_1$ because it has a higher average reward.

(b) The agent should choose $a_2$ because it has a higher average reward.

(c) The agent cannot determine which action to take because it does not have enough information about the environment.

(d) The agent should choose $a_1$ because it has a higher maximum reward.