UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
CS 440/ECE 448 Artificial Intelligence
Spring 2019

EXAM 1 REVIEW

Tuesday, February 26, 2019

- This is a CLOSED BOOK exam. No calculators are permitted. You need not simplify explicit numerical expressions.
- This review includes many problems. Exam is typically 8 T/F questions, 8 short answer questions, and 3 long questions.

Name: ______________________

netid: ______________________

Section: ______________________

1. (2 points) An unknown environment is also unobservable.

   ◯ True

   √ **False**

   Explain:

   Solution: It may be possible to know the entire current state of the world, and yet not know the result of an action.

2. (2 points) In a fully observable, known, static, single-agent, sequential, stochastic environment, an internal-state agent has no more information available to it than a reflex agent.

   √ **True**

   ◯ False

   Explain:

   Solution: An internal-state agent has an internal model of the world. The internal state is useful if the environment is partially observable (because it estimates the unobserved information), unknown (because it can estimate the unknown transition model), dynamic (because it can keep track of time elapsed while making a decision), or multi-agent (because it can estimate the other player's strategy). If the environment is fully observable, known, static and single-agent, then the internal state is not really useful for anything. The fact that the environment is sequential doesn't matter if it is fully observable, because both agents know where they are. The fact that the environment is stochastic doesn't matter if it is known, because both agents know the transition probabilities.

3. (2 points) Suppose you are given a "perfect" heuristic function that gives the correct minimum cost from each node to the goal. True or false: greedy best-first search with this heuristic always returns the solution with the lowest cost.

   ◯ True

   √ **False**

   Explain:

   Solution: Suppose the search contains four nodes: Start, A, B, and Goal, with the following distances:

$$d(S,A) = 10$$
$$d(S,B) = 5$$
$$d(A,G) = 1$$
$$d(B,G) = 2$$

   Greedy search chooses node A, because it has the lowest remaining distance to goal, even though node B is on the shortest path.

4. (2 points) The GRAPHPLAN algorithm develops a planning graph until it reaches a level at which no pair of goal predicates is mutex, and then stops.

   √ **True**

◯ False

Explain:

Solution: GRAPHPLAN is a lower bound on the cost-to-goal. It is computed by populating each level of a planning graph from the previous level, until a level is reached at which all goal predicates exist, and no two of them are mutually exclusive (mutex).

5. (2½ points) Give one reason why a test of intelligence should evaluate action rather than thought. Give one reason why such a test should evaluate thought rather than action.

Solution:

- One reason to evaluate action: it is difficult to evaluate the thought processes of a human being, so a test of thought can't really evaluate the intelligence of human beings.
- One reason to evaluate thought: in a closed environment, a simple reflex (rule-based) software agent can easily be programmed to act with perfect rationality (i.e., in a way that optimally maximizes its own reward, even if it has no internal model of "reward" or any kind of self-awareness).

6. (2½ points) Can an environment be both known and unobservable? Give an example.

Solution: An environment can be both known and unobservable. An example of this is Russian roulette. Russian roulette is a potentially lethal game of chance in which a player places a single round in a revolver, spins the cylinder, places the muzzle against his or her head, and pulls the trigger1. The environment is known beforehand (i.e., each agent continues on until the other agent is terminated from the game) but the revolver is unobservable from agents (they cannot observe where the bullet is located).

7. (2½ points) What is the difference between a goal-directed agent and a utility-directed agent? Given an example of each, from among the software agents that you have written for machine problems in this course.

Solution: A goal-directed agent works toward a future goal: an example is the pentomino agent from MP2. A utility-directed agent works to maximize a measure of utility: examples include all of the minimax and alpha-beta agents from MP2.

8. (2½ points) Discuss the relative strengths and weaknesses of breadth-first search vs. depth-first search for AI problems.

Solution:

	BFS	DFS
Strength	Solution is guaranteed to be optimal. BFS is complete.	Can find goal faster than BFS if there are multiple solutions. Linear memory requirement ($O(bd)$).
Weakness	Exponential ($O(b^d)$) space complexity.	Solution not guaranteed to be optimal. Not complete. Computation is exponential in longest path ($O(b^m)$), rather than shortest path ($O(b^d)$).

9. ($2\frac{1}{2}$ points) In the tree search formulation, why do we restrict step costs to be non-negative?

> **Solution:** If the cost around any loop is negative, then the lowest-cost path is to take that loop an infinite number of times.

10. ($2\frac{1}{2}$ points) What is the distinction between a world state and a search tree node?

> **Solution:** A world state contains enough information to know (1) whether or not you've reached the goal, (2) what actions can be performed, (3) what will be the result of each action. A search tree node contains a pointer to the world state, plus a pointer to the parent node.

11. ($2\frac{1}{2}$ points) How do we avoid repeated states during tree search?

> **Solution:** By keeping a set of "explored states." If expanding a search node results in a state that has already been explored, we don't add it to the frontier.

12. ($2\frac{1}{2}$ points) How can randomness be incorporated into a game tree? How about partial observability (imperfect information)?

> **Solution:** Randomness is incorporated using the expectiminimax algorithm, in which max tries to maximize the expected score, min tries to minimize the expected score. Partial observability is incorporated using a minimax state tree in which neither player knows for sure which state they're in; the max player chooses an action that maximizes the minimum payoff over all of the states he might be in.

13. ($2\frac{1}{2}$ points) What are the main challenges of adversarial search as contrasted with single-agent search? What are some algorithmic similarities and differences?

> **Solution:** The biggest difference is that we are unaware of how the opponent(s) will act. Because of this our search cannot simply consider my own moves, it must also figure out how my opponent will act at each level, thus effectively doubling the number of levels over which I have to search. Since the number of levels is the exponent in the computational complexity, this makes computational complexity much harder.

14. ($2\frac{1}{2}$ points) What additional difficulties does dice throwing or other sources of uncertainty introduce into a game?

> **Solution:** Uncertainties introduce probabilities into the game. Expectiminimax is used to find solutions for these type of games. Expectiminimax doubles the number of levels as compared to minimax, because there is randomness after every play. Expectiminimax has nasty branching factor and often times defining evaluation functions and pruning algorithms are difficult.

15. ($2\frac{1}{2}$ points) What is local search for CSPs? For which kinds of CSPs might local search be better than backtracking search? What about the other way around?

> **Solution:** Local search starts with all variables assigned, then changes one variable at a time, attempting to re-assign the variable in a way that eliminates constraint violations. Local search makes sense if the problem has few bad local optima (a bad local optimum is a complete variable assignment that doesn't satisfy all the constraints, but from which any change of any variable will increase the number of violated constraints).

16. ($2\frac{1}{2}$ points) 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.

> **Solution:**
>
> - When we choose a variable, we are not eliminating any possible solutions, we are only deciding on the order in which variables will be considered. It makes sense to choose the order that minimizes complexity.
> - When we choose an assignment, we are eliminating possible solutions; if we're wrong, we'll have to back-track. Therefore it makes sense to choose the assignment that eliminates as few solutions as possible, to minimize the chance of back-tracking.

17. ($2\frac{1}{2}$ points) Use the axioms of probability to prove that $P(\neg A) = 1 - P(A)$.

> **Solution:**
>
> - From the third axiom, $P(A \vee \neg A) = P(A) + P(\neg A) - P(A \wedge \neg A)$.
> - The event $(A \vee \neg A)$ is always true, so from the second axiom, $P(A \vee \neg A) = 1$. The event $(A \wedge \neg A)$ is always false, so from the second axiom, $P(A \wedge \neg A) = 0$.
> - Combining the two statements above, $1 = P(A) + P(\neg A)$. Q.E.D.

18. ($2\frac{1}{2}$ points) Consider the following joint probability distribution:

$$
\begin{aligned}
P(A, B) &= 0.12 \\
P(A, \neg B) &= 0.18 \\
P(\neg A, B) &= 0.28 \\
P(\neg A, \neg B) &= 0.42
\end{aligned}
$$

What are the marginal distributions of A and B? Are A and B independent and why?

> **Solution:** $P(A) = 0.3, P(\neg A) = 0.7, P(B) = 0.4, P(\neg B) = 0.6$. They are independent, because $P(A)P(B) = P(A, B) = 0.12, P(A)P(\neg B) = P(A, \neg B) = 0.18$, and so on.

19. ($2\frac{1}{2}$ points) 20% of students at U of I are part of the Greek system. Amongst these students, 10% study engineering. Furthermore, 15% of the entire student body studies engineering. Given that we know that a student studies engineering, what is the probability that the student is not part of the Greek system?

Solution: Define G=student part of the Greek system, E=student studies engineering. We are given that $P(G) = 0.2$ and $P(E|G) = 0.1$, from which we may infer that $P(E, G) = 0.02$. We are also given that $P(E) = 0.15$, from which we may infer that

$$P(\neg G, E) = P(E) - P(E, G) = 0.13$$
$$P(\neg G|E) = \frac{P(\neg G, E)}{P(E)}$$
$$= \frac{0.13}{0.15} = \frac{13}{15}$$

20. (2½ points) Give an example of a coordination game and an anti-coordination game. For each game, write down its payoff matrix, list dominant strategies, Pareto optimal solutions, and pure strategy Nash equilibria (if any).

Solution: The stag hunt is a coordination game. The payoff matrix is

	Player 1: Cooperate	Player 1: Defect
Player 1: Cooperate	2, 2	0,1
Player 2: Defect	1,0	1,1

There are no dominant strategies. The pure-strategy Nash equilibria are (Cooperate,Cooperate) and (Defect,Defect). The Pareto-optimal solution is (Cooperate,Cooperate).

The game of Chicken is an anti-coordination game. The payoff matrix is

	Player 1: Chicken	Player 1: Drive
Player 1: Chicken	0, 0	-1,1
Player 2: Drive	1,-1	-10,-10

There are no dominant strategies. The Pareto optimal solutions are (Chicken,Chicken), (Chicken,Drive), and (Drive,Chicken). The pure-strategy Nash equlibria are (Chicken,Drive) and (Drive,Chicken).

21. (2½ points) In the lectures, we covered dominant strategies, Nash equilibria, and Pareto optimal solutions of simultaneous move games. We can also consider minimax strategies for such games, defined in the same way as for multi-player alternating games. What would be the minimax strategies in the Prisoners Dilemma, Stag Hunt, and Game of Chicken? If both players follow the minimax strategy, does the game outcome differ from the Nash equilibria? When/why would one prefer to choose a minimax strategy rather than a Nash equilibrium?

Solution: Minimax solution maximizes, over all of your possible actions, the minimum, over all of your opponent's possible actions, of your reward.

- Prisoner's Dilemma: Defect. Result is also the Nash equilibrium.
- Stag Hunt: take the Hare. Result is one of the two Nash equilibria.
- Game of Chicken: chicken out. Result is not a Nash equilibrium.

Nash equilibrium assumes that you know what the other player will do, and can respond appropriately. Minimax makes more sense if you want to limit your losses, and have no way to predict the other player's behavior.

22. (7 points) Suppose you have to design an agent to play the game of Scrabble. Write a PEAS specification for the agent and characterize the task environment according to the seven relevant categories.

Solution: Performance Measures:

- maximizing own score
- minimizing opponents' scores

Environment:

- board squares
- placed tiles
- personal set of tiles

Actuators:

- place tile
- select new tiles

Sensors:

- see each piece on the board

Environment:

- Partially Observable
- Strategic
- Sequential
- Static
- Discrete
- Multi-Agent
- Known

23. (6 points) For each type of maze described below, specify the typical-case time complexity and space complexity of both breadth-first-search (BFS) and depth-first-search (DFS). Assume that both BFS and DFS return the first solution path they find.

   (a) The Albuquerque maze has $b = 3$ possible directions that you can take at each intersection. No path is longer than $m = 25$ steps. There is only one solution, which is known to require exactly $d = 25$ steps.

> **Solution:** Both BFS and DFS are $O\left\{b^m\right\} = O\left\{b^d\right\}$.

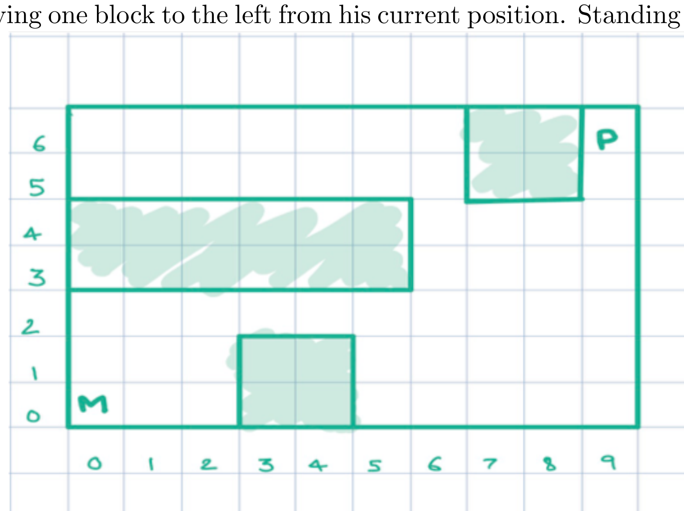
(b) The Belmont maze has $b = 3$ possible directions that you can take at each intersection. No path is longer than $m = 25$ steps. About half of all available paths are considered solutions to the maze.

> **Solution:** BFS is $O\left\{b^m\right\}$. DFS has an expected computational cost of $O\left\{m\right\}$ (expected value of the path cost turns out to be exactly $2m$, if you work it out).

(c) The Crazytown maze has $b = 3$ possible directions that you can take at each intersection. The maze is infinite in size, so some paths have infinite length. There is only one solution, which is known to require $d = 25$ steps.

> **Solution:** BFS is $O\left\{b^d\right\}$. DFS is $O\left\{b^\infty\right\}$.

24. (8 points) Refer to the maze shown below. Here, M represents Mario, P represents Peach, and the goal of the game is to get Mario and Peach to find each other. In each move, both Mario and Peach take turns. For example, one move would consist of Peach moving a block to the bottom from her current position, and Mario moving one block to the left from his current position. Standing still is also an option.



(a) Describe state and action representations for this problem.

> **Solution:**
>
> - State = (x_M, y_M) position of Mario, (x_P, y_P) position of Peach.
> - Action = Mario moves up,down,left,right, or stationary, Peach moves up,down,left,right, or stationary.

(b) What is the branching factor of the search tree?

> **Solution:** 25

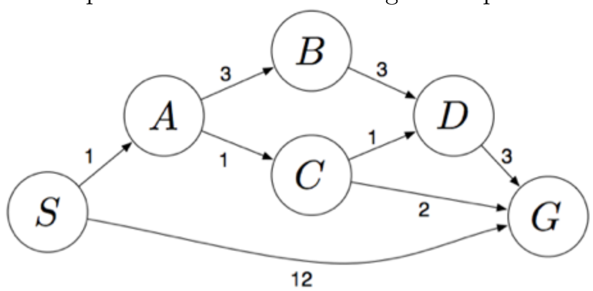
(c) What is the size of the state space?

Solution: There are 50 possible locations for both Mario and Peach (if we assume that they can occupy the same square), for a total state space of 2500.

(d) Describe an admissible heuristic for this problem.

Solution: The heuristic $h(n) = 0$ is always admissible, but would receive at most a little bit of partial credit, because it's trivial. A more useful heuristic would measure the distance between Mario and Peach, e.g., $|x_M - x_P| + |y_M - y_P|$.

25. (8 points) Consider the search problem with the following state space:



S denotes the start state, G denotes the goal state, and step costs are written next to each arc. Assume that ties are broken alphabetically (i.e., if there are two states with equal priority on the frontier, the state that comes first alphabetically should be visited first).

(a) What path would BFS return for this problem?

Solution: SG

(b) What path would DFS return for this problem?

Solution: SABDG

(c) What path would UCS return for this problem?

Solution: SACG

(d) Consider the heuristics for this problem shown in the table below.

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

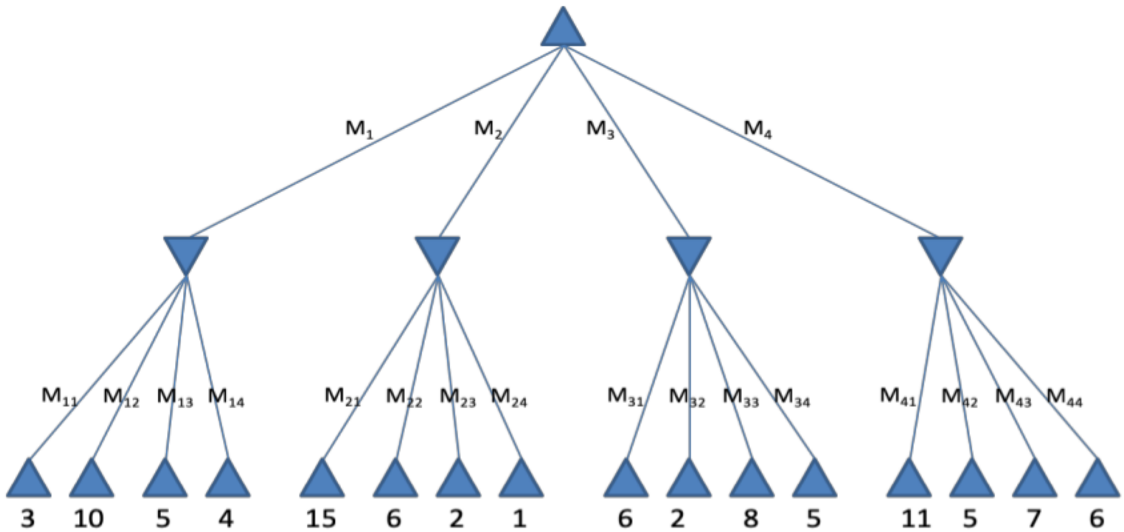
i. Is h1 admissible? Is it consistent?

Solution: Neither admissible nor consistent.

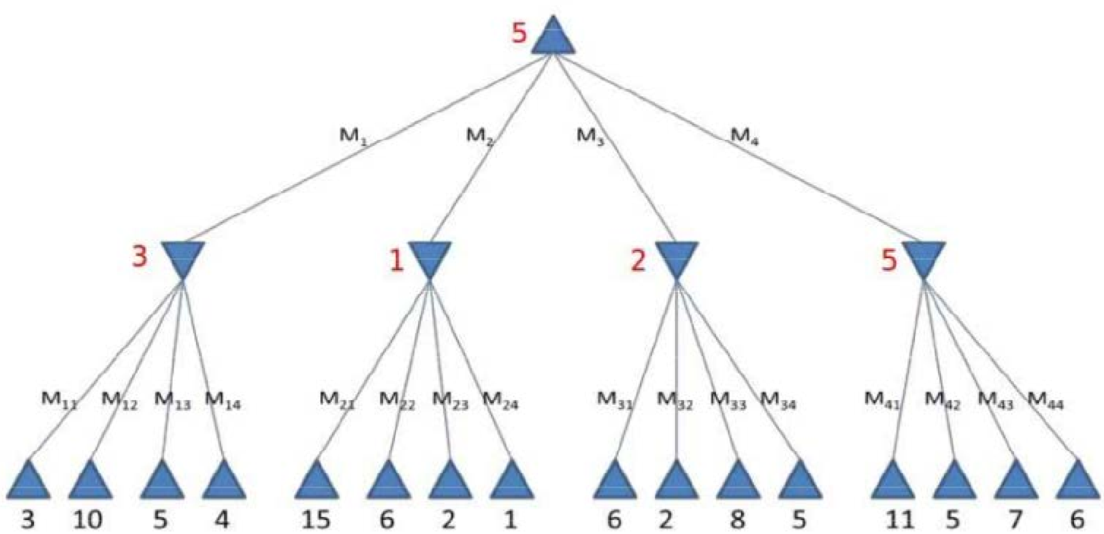
ii. Is h2 admissible? Is it consistent?

> **Solution:** Admissible but not consistent.

26. (8 points) Consider the following game tree (MAX moves first):



(a) Write down the minimax value of every non-terminal node next to that node.

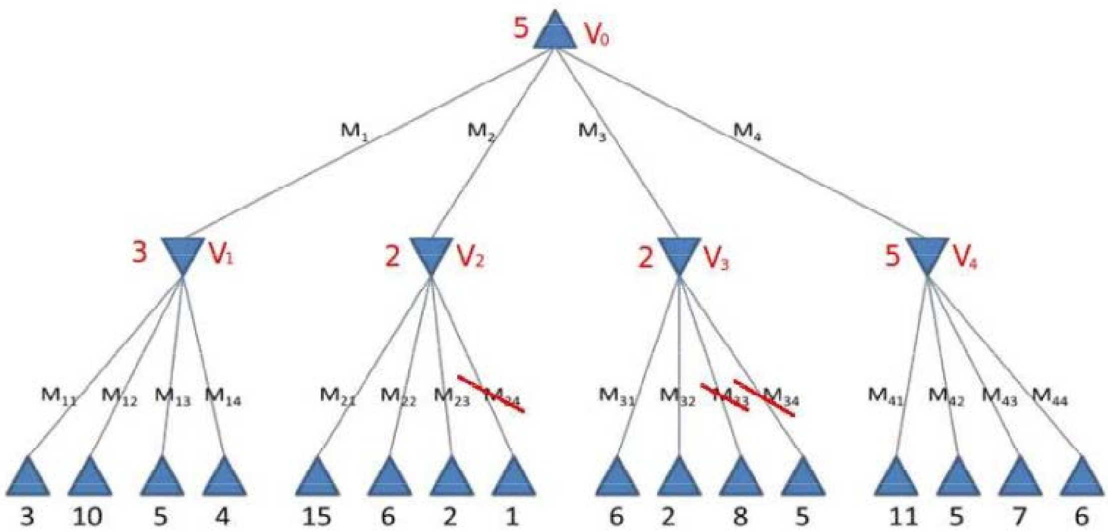
> **Solution:**
>
> 
>

(b) How will the game proceed, assuming both players play optimally?

> **Solution:** The game will choose the max value on depth 1, taking route M_4. It will then take the minimum value on depth 2 that is child of the chosen node, and hence take M_{42}.

(c) Cross out the branches that do not need to be examined by alpha-beta search in order to find the minimax value of the top node, assuming that moves are considered in the non-optimal order shown.

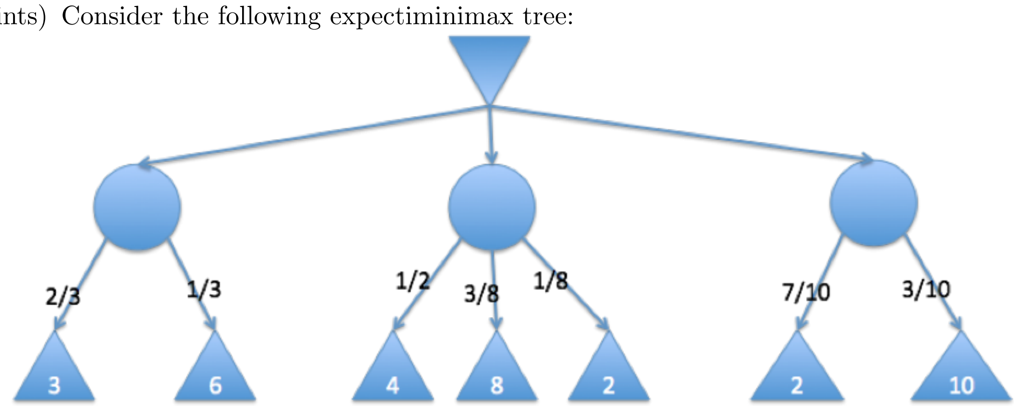
Solution:



(d) Suppose that a heuristic was available that could re-order the moves of both max (M_1, M_2, M_3, M_4) and min $(M_{11}, \ldots, M_{44})$ in order to force the alpha-beta algorithm to prune as many nodes as possible. Which max move would be considered first: M_1, M_2, M_3, or M_4? Which of the min moves $(M_{11}, \ldots, M_{44})$ would have to be considered?

Solution: The first max move to be considered would be M_4, because it allows us to set the highest α. Only 7 of the min moves would be considered: M_{41} through M_{44} would have to be considered to determine that $\alpha = 5$, and then (if the heuristic magically sorts moves in order for us), we would consider M_{32}, M_{24}, and M_{11}, find that all of them have values below α, and prune away their parents.

27. (8 points) Consider the following expectiminimax tree:



Circle nodes are chance nodes, the top node is a min node, and the bottom nodes are max nodes.

(a) For each circle, calculate the node values, as per expectiminimax definition.

Solution: From left to right: 4, 5.25, 4.4.

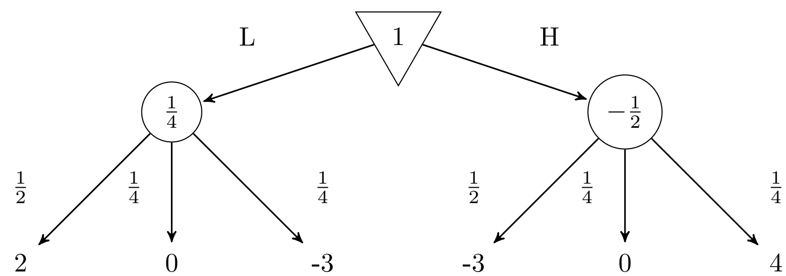
(b) Which action should the min player take?

Solution: The first action.

28. (8 points) Consider the following game, called "High/Low." There is an infinite deck of cards, half of which are 2s, one quarter are 3s, and one quarter are 4s. The game starts with a 3 showing. After each card, you say "High" or "Low," and a new card is flipped. If you are correct (e.g., you say "High" and then the next card is higher than the one showing), you win the points shown on the new card. If there is a tie (the next card equals the one showing), you get zero points. If you are wrong (e.g., you say "High" and then the next card is lower than the one showing), then you lose the amount of the card that was already showing.

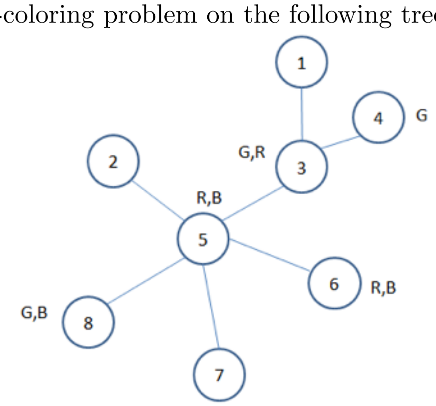
    Draw the expectimax tree for the first round of this game and write down the expected utility of every node. What is the optimal policy assuming the game only lasts one round?

Solution:



Optimal policy: L

29. (6 points) Consider the graph-coloring problem on the following tree-structured CSP:



We assume there are three available colors (R,G,B) and some nodes are constrained to use only a subset of these colors, as indicated above. Show all the steps for applying the tree-structured CSP algorithm for finding a solution to this problem.

Solution:

1. Sort the nodes into a search order with no crossed edges, e.g., 2,8,7,6,5,3,1,4.
2. Perform arc consistency on each pair of nodes, from right to left. With the ordering specified above, the following changes are made: 3 cannot be G. 5 cannot be R. 2, 8, 7, 6 cannot be B.

3. Assign the variables to any remaining value, from left to right. For example, 2=G, 8=G, 7=G, 6=R, 5=B, 3=R, 1=B, 4=G.

30. (5 points) For each of the following problems, determine whether an algorithm to optimally solve the problem requires worst-case computation time that is polynomial or exponential in the parameters d and m (assuming that P $\neq$ NP).

(a) A map has d regions. Colors have been applied to all m regions, drawing from a set of m possible colors. Your algorithm needs to decide whether or not any two adjacent regions have the same color.

Solution: Polynomial in d and m.

(b) b. A map has d regions. Your algorithm needs to assign colors to all d regions, drawing colors from a set of m possible colors, in order to guarantee that no two adjacent regions have the same color.

Solution: Exponential in d.

(c) Your algorithm needs to find its way out of a maze drawn on a d-by-d grid.

Solution: Polynomial in d.

(d) Your algorithm needs to find the shortest path in a d-by-d maze while hitting m waypoints (equivalent to dots in MP1 part 1.2).

Solution: Exponential in m.

(e) Your algorithm needs to solve a planning problem in a blocks world consisting of d blocks.

Solution: Exponential in d

31. (7 points) Let A and B be independent binary random variables with $p(A = 1) = 0.1$, $p(B = 1) = 0.4$. Let C denote the event that at least one of them is 1, and let D denote the event that exactly one of them is 1.

(a) What is $P(C)$?

Solution:

$$\begin{aligned} P(C) &= p(A = 1, B = 1) + p(A = 1, B = 0) + p(A = 0, B = 1) \\ &= (0.1)(0.4) + (0.1)(0.6) + (0.9)(0.4) = 0.46 \end{aligned}$$

where the last line follows from the independence of A and B.

(b) What is $P(D)$?

Solution:

$$\begin{aligned} P(D) &= p(A = 1, B = 0) + p(A = 0, B = 1) \\ &= (0.1)(0.6) + (0.9)(0.4) = 0.42 \end{aligned}$$

(c) What is $P(D|A=1)$?

Solution:

$$\begin{aligned} P(D|A=1) &= P(D, A=1)/p(A=1) \\ &= p(A=1, B=0)/p(A=1) \\ &= \frac{0.06}{0.1} = 0.6 \end{aligned}$$

(d) Are A and D independent? Why?

Solution: No. $P(D|A=1) \neq P(D)$.

32. (12 points) Consider the following game:

	Player A: Action 1	Player A: Action 2
Player B: Action 1	A=3 B=2	A=0 B=0
Player B: Action 2	A=1 B=1	A=2 B=3

(a) Find dominant strategies (if any).

Solution: A dominant strategy is defined as a strategy whose outcome is better for the player regardless of the strategy chosen by the other player. Let's first look for dominant strategies for A: Suppose B chooses Action1. A gets 3 if it chooses Action1 or 0 if it chooses Action2. So it shoould choose Action1. Now suppose B chooses Action2. A gets 1 if it chooses Action1 or 2 if it chooses Action2. So it should choose Action2. Thus there is no dominant strategy for A. Let's look at B: Suppose A chooses Action1. B gets 2 if it chooses Action1 or 1 if it chooses Action2. So it should choose Action1. Now suppose A chooses Action2. B gets 0 if it chooses Action1 or 3 if it chooses Action2. So it should choose Action2. Thus there is also no dominant strategy for B.

(b) Find pure strategy equilibria (if any).

Solution: A Nash Equilibrium is a set of strategies such that no player can get a bigger payoff by switching strageties, provided the other player sticks with the same strategy. There are two: (A: Action1, B: Action1) or (A: Action2, B: Action2).

33. (12 points) In each square, the first number refers to payoff for the player whose moves are shown on the row-label, the second number refers to payoff for the player shown on the column label.

	A	B	C
A	0, 0	25, 40	5, 10
B	40, 25	0, 0	5, 15
C	10, 5	15, 5	10, 10

(a) Are there any dominant strategies? If so, what are they? If not, why not?

Solution: No. The best move, for each player, depends on what the other player does.

(b) Are there any pure-strategy Nash equilibria? If so, what are they? If not, why not?

Solution: (A',B), (A,B')

(c) Are there any Pareto-optimal solutions? If so, what are they? If not, why not?

Solution: (A',B) and (A,B').

34. (12 points) Suppose that both Alice and Bob want to go from one place to another. There are two routes R1 and R2. The utility of a route is inversely proportional to the number of cars on the road. For instance, if both Alice and Bob choose route R1, the utility of R1 for each of them is 1/2.

(a) Write out the payoff matrix.

Solution:

	Alice R1	Alice R2
Bob R1	A:0.5, B:0.5	A:1, B:1
Bob R2	A:1, B:1	A:0.5, B:0.5

(b) Is this a zero-sum game?

Solution: No.

(c) Find dominant strategies, if any. If there are no dominant solution, explain why not.

Solution: There is no dominant solution. The best strategy for each player depends on the strategy of the other player.

(d) Find pure strategy equilibria, if any. If there are no pure strategy equilibria, explain why not.

Solution: There are two: (Alice=R1,Bob=R2) and (Alice=R2,Bob=R1).

(e) Find the mixed strategy equilibrium.

Solution: Alice chooses R1 with probability p, and R2 with probability $1-p$. p must be chosen so that Bob's reward is independent of the action he takes.

- Bob's Reward(R1)$= 0.5p + (1-p) = 1 - 0.5p$
- Bob's Reward(R2)$= p + 0.5(1-p) = 0.5 + 0.5p$

Setting the two rewards equal, we find $p = 0.5$.