DQ7 (L7)

Due 12 Mar at 23:59 **Points** 20 **Questions** 6
Available after 6 Mar at 12:00 **Time limit** None
Allowed attempts Unlimited

Instructions

- This quiz is NOT GRADED. However, it is HIGHLY RECOMMENDED that you use these questions to complement your review of the lecture content.
- The questions are based on content from the Lecture 7 and from part of Chapter 6 of the AIMA (4th Ed.) textbook (i.e., 6.1-6.3).

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	7 minutes	20 out of 20
LATEST	Attempt 2	7 minutes	20 out of 20
	Attempt 1	7 minutes	8 out of 20

Submitted 6 Mar at 12:14

Question 1

2 / 2 pts

Which of the following is **NOT** part of a problem formulation for adversarial search?

- [ ] TO-MOVE(s)
- [ ] ACTIONS(s)
- [x] GOAL-TEST(s) Correct!
- [ ] UTILITY(s, p)

With adversarial search, instead of GOAL-TEST(s), we have IS-TERMINAL(s).

IS-TERMINAL(s) is a terminal test. It returns true when the game is over and false otherwise. States where the game has ended are called terminal states.

Question 2

2 / 2 pts

What is the definition of a winning strategy?

- A winning strategy for player 1, s_1^*, implies that, for any strategy used by player 2, s_2, the game ends in either a tie or win for player 1.
- A winning strategy for player 1, s_1^*, implies that, for some strategy used by player 2, s_2, the game ends in a win for player 1.
- A winning strategy for player 1, s_1^*, implies that, for some strategy used by player 2, s_2, the game ends in either a tie or win for player 1.
- Correct! A winning strategy for player 1, s_1^*, implies that, for any strategy used by player 2, s_2, the game ends in a win for player 1.

As defined (see correct answer).

Question 3

4 / 4 pts

Which of the following properties of the Minimax algorithm is true?

Assume that all payoffs are stated relative to the MAX player's utility.

Select the best option.

- Minimax chooses moves for the MAX player that maximise across MIN player's payoffs, while choosing moves for the MIN player that minimise across the MAX player’s payoffs.
- The Minimax algorithm is similar to Breath-First Search (BFS) in terms of the order of nodes explored.
- All of the above mentioned.
- None of the above mentioned.

Correct!

For Option 1:

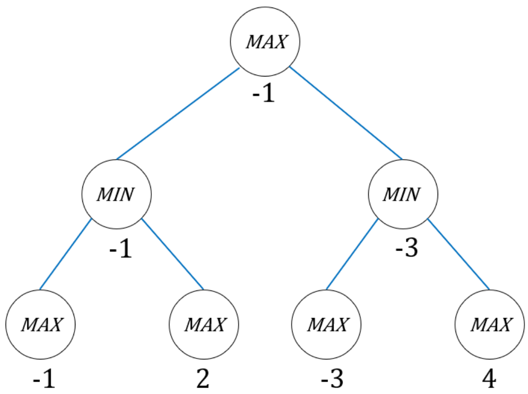
$$Minimax(s) = \begin{cases} Utility(s) \text{ if TerminalTest}(s) \\ \max_{a \in \text{Actions}(s)} \text{Minimax}(\text{Result}(s,a)) \text{ if Player}(s) = \text{MAX} \\ \min_{a \in \text{Actions}(s)} \text{Minimax}(\text{Result}(s,a)) \text{ if Player}(s) = \text{MIN} \end{cases}$$

This equates to:

- MAX chooses the move that maximises MAX player utility; this is the same as minimising MIN player utility.
- MIN chooses move to minimises MAX player utility

Refer to the following example for further clarity:

We can see that MAX will maximise from among the utility values at the next level (e.g., maximise between the -1 and -3 values). However, since the MIN player's payoff or utility are actually 1 and 3 respectively, we notice that MAX is actually minimising the MIN player's utility (i.e., picking the action that yields a MIN player utility of 1 (instead of 3). Interpreting that MAX maximises only works when we refer to utility values in terms of the MAX player only. However, the MIN player does pick an action that minimises the MAX player's utility values (notice here that we are using MAX player utility values any way).



For Option 2:

Minimax performs a complete Depth-First exploration of the game tree.

Question 4

4 / 4 pts

Which of the following statements regarding the Minimax algorithm is true?

Select the one or more options that are true.

Correct!

- [x] The utility (i.e., utility in terms of the MAX player's perspective) obtained by MAX using minimax decisions against a suboptimal MIN (i.e., a MIN that does not choose all the best options for MIN) will never be lower than the utility (i.e., utility in terms of the MAX player's perspective) obtained playing against an optimal MIN.

- [ ] The utility (i.e., utility in terms of the MAX player's perspective) obtained by MAX using minimax decisions against a suboptimal MIN (i.e., a MIN that does not choose all the best options for MIN) will still be optimal for MAX (i.e., highest possible utility within the subtree, where utility is in terms of the MAX player's perspective).

Correct!

- [x] The utility (i.e., utility in terms of the MAX player's perspective) obtained by MIN using minimax decisions against a suboptimal MAX (i.e., a MAX that does not choose all the best options for MAX) will never be higher than the utility (i.e., utility in terms of the MAX player's perspective) obtained playing an optimal MAX.

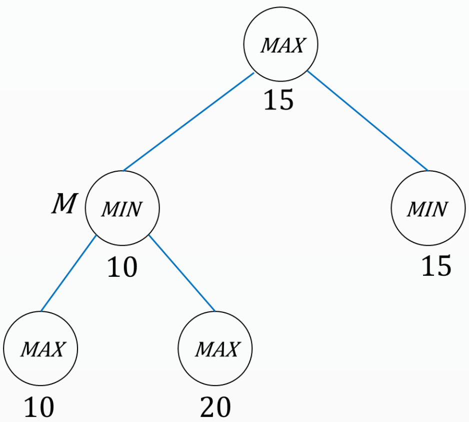
- [ ] The utility (i.e., utility in terms of the MAX player's perspective) obtained by MIN using minimax decisions against a suboptimal MAX (i.e., a MAX that does not choose all the best options for MAX) will still be optimal for MIN (i.e., lowest possible utility in the subtree, where utility is in terms of the MAX player's perspective).

- **Option 1**: True
  Consider a MIN node, **M**, whose children are terminal nodes. If MIN plays suboptimally, then the utility (i.e., from MAX perspective) value at **M** must be greater than or equal to the utility (i.e., from MAX perspective) value under an optimal MIN player (since the optimal MIN player would pick the lowest utility at **M**). Hence, the value of the MAX node that is the parent of **M** can only be increased. This argument can be extended by a simple induction all the way to the root.

- **Option 2**: False

  Consider the following game tree. Under Minimax, the utility for MAX (at the root) is 15 since under the Minimax strategy, we know that MIN would play optimally at node *M*, picking the action that yields a utility of 10.

  However, in the case where MIN plays sub-optimally, then at *M*, MIN might pick the action which yields utility 20. In this case, MAX (planning using Minimax) would not have picked the optimal move.



- **Option 3**: True
  Consider a MAX node, **X**, whose children are terminal nodes. If MAX plays suboptimally, then the utility (i.e., from MAX perspective) value at **X** must be less than or equal to the utility (i.e., from MAX perspective) value under an optimal MAX player (since the optimal MAX player would pick the highest utility at **X**). Hence, the value of the MIN node that is the parent of **X** can only be decreased. This argument can be extended by a simple induction all the way to the root.

- **Option 4**: False

  Consider the following game tree. Under Minimax, the utility for MIN (at the root) is 15 since under the Minimax

Question 5

4 / 4 pts

Which of the following options are true about the alpha-beta pruning algorithm?

Select one or more options. Only pick the last option (i.e., none of the above) if none of the other 3 options are true.

- [ ] Alpha corresponds to the MIN value whereas Beta corresponds to the MAX value.
- [ ] Pruning a node using the alpha-beta pruning algorithm may affect the outcome of the search.

Correct!

- [x] Given a MIN node, ***n***, stop searching below ***n*** if there is some MAX ancestor ***i*** with α(***i***) > β(***n***).
- [ ] None of the above.

Option 1: Alpha corresponds to the **MAX** value whereas Beta corresponds to the **MIN** value.

Option 2: The alpha-beta pruning algorithm does not affect outcome of the game. It only seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree by pruning off nodes/ sub-trees that are guaranteed to give a worse utility for the MAX player.

Option 3: Given a MIN node ***n***, stop searching below ***n*** if there is some MAX ancestor ***i*** with $\alpha(i) \geq \beta(n)$. Take note that if $\alpha(i) = \beta(n)$, pruning **will** occur as well. However, the given statement still holds despite this since the algorithm does prune the values that satisfy $\alpha(i) > \beta(n)$ (i.e., the statement is vacuously true for the cases where $\alpha(i) = \beta(n)$). The proper false statement would instead be as follows. "Given a MIN node ***n***, stop searching below ***n*** **if and only if** there is some MAX ancestor ***i*** with $\alpha(i) > \beta(n)$."

Question 6

4 / 4 pts

Suppose you were allowed to adjust the game tree using the following options.

Select the options that would allow one to maximise the effectiveness of the alpha-beta pruning algorithm (i.e., result in the most pruning).

Correct!

- [x] Picking the ordering over the branching at each node.
- [ ] Chaning the utility values at the lead nodes such that they are all equal.
- [ ] Fixing the depth of the game tree (i.e., defining all nodes at the chosen depth as leaf nodes, setting their utility values based on the Minimax strategy).
- [ ] Switching the order of MIN and MAX players.

Recall that the purpose of the alpha-beta pruning is to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree.

Option 1: Move ordering is an important aspect of alpha-beta pruning. The ideal ordering ordering would result in more pruning.

Option 2: Notice that when all of the utility values of the leaf nodes are equal, we can easily fulfil the condition of pruning. For a MIN node, that is when its **beta value** is less than or **equals to** (≤) the **alpha value** of the MAX ancestor. For a MAX node, that is when its **alpha value is** greater than or **equal to** (≥) the **beta value** of the **MIN** ancestor. Hence, with all leaf nodes having the same utility, we are assured of maximum pruning, maximising the effectiveness of the alpha-beat pruning algorithm.

Option 3 and 4: The depth of the game tree (Option 3) and the order in which MIN or MAX starts first (Option) does not affect the effectiveness of alpha-beta pruning