Introduction to Artificial Intelligence

UNIZG FER, AY 2021/2022

Exercises, v1

4 Game playing

1 (T) Both heuristic seach algorithms and the minimax algorithm make use of a heuristic function. In both cases the function serves to compute an estimate of the quality of the current state. However, there's a difference in how the heuristic function is used in the two cases. **What is the difference between how the heuristic function is used in heuristic search algorithms and how it's used in the minimax algorithm?**

- A Minimax does a depth-first search, so it always applies the heuristic function on the node at the deepest level, whereas a heuristic search algorithm applies the heuristic on the first node in the list of open nodes
- B In minimax, the heuristic is not applied on all nodes in the game tree but only on nodes at a certain depth, whereas in heuristic search algorithms the heuristic is applied on all the generated nodes
- C A heuristic search algorithm always expands the node with the lowest heuristic value, as this state is the closest to the goal state, whereas minimax will always select the node with the largest heuristic value
- D Heuristic search algorithms sort the list of open nodes based on the value of the heuristic, whereas minimax always extends the node with the largest heuristic value, thereby effectively doing a depth-first search

2 (C) A game tree is defined by the following transitions: $succ(A) = \{B, C\}, succ(B) = \{D, E\}, succ(D) = \{H, I\}, succ(E) = \{J, K, L\}$,$succ(C) = \{F, G\}, succ(F) = \{M, N, O\}, succ(G) = \{P, Q\}$. The heuristic values of the leaf nodes are the following: $h(H) = h(J) = 8, h(I) = -15, h(K) = -h(Q) = 17, h(L) = h(O) = 1, h(M) = -12, h(N) = -19, h(P) = -14$. The optimal strategy is determined using the minimax algorithm with alpha-beta pruning. The first move is made by the MAX player. In doing so, **which nodes will be pruned away (skipped in minimax computation)**?

A K, L, G, P, Q B L, Q C O, P, Q D L, F, M, N, O

3 (C) Consider a two-player zero-sum game. Every state $s \in S$ of this game can be compactly encoded as a 3-digit natural number between 111 and 999. The successor states of s are defined as all states that can be obtained from s by incrementing one digit by one, e.g., $succ(235) = \{335, 245, 236\}$. However, the states that contain the digit 9 are terminal states and as such have no successor states, e.g., $succ(932) = \varnothing$. In terminal states, the payoff for the first player (MAX) equals the difference between the first and the third digit, e.g., $utility(932) = 9 - 2 = 7$. The payoff for the second player (MIN) is the negative of the payoff for the first player. The game is being played by two minimax algorithms, A_1 (MAX player) and A_2 (MIN player). Both search up to two moves ahead, i.e., the depth limit of minimax is set to 2. However, the algorithms use different heuristics. Heuristic h_1 used by A_1 returns the first digit from s, while heuristic h_2 used by A_2 (as defined from the viewpoint of that algorithm) returns the third digit from s. E.g., $h_1(236) = 2$ and $h_2(236) = 6$ (when computing the minimax values in a game tree rooted in MAX, you'll have to negate the values of h_2). Let the initial game state be $s_0 = 175$. Algorithm A_1 (MAX player) is to make the first move. **Through what sequence of states will the game evolve, if both players employ the minimax strategy?**

A $175 \to 275 \to 375$ B $175 \to 176 \to 186$ C $175 \to 275 \to 276$ D $175 \to 176 \to 276$

4 (P) Consider two game playing algorithms: one with heuristic h_1 and search depth d_1, and another with heuristic h_2 and search depth d_2. Both algorithms use the minimax strategy. **When will the first algorithm more often win over the second algorithm?**

A If $h_1 > h_2$ B If $d_1 = d_2$ and $h_1 > h_2$ C If $h_1 = h_2$ and $d_1 > d_2$ D If $d_2 < d_1$

5 (P) Four game playing algorithms, A_1, A_2, A_3, and A_4, are competing in a chess tournament. All four algorithms are implemented as minimax with alpha-beta pruning. However, they differ in what heuristic and search depth limit they use. Let h_i and d_i be the heuristic and the depth limit of algorithm A_i, respectively. Shortly before the tournament, some key information was leaked to the public: (1) h_3 is the best of the four heuristics, (2) h_2 and h_4 are identical, (3) due to an error, A_3 has alpha-beta pruning switched off, and (4) the depth limits are such that $d_1 > d_2 = d_3 > d_4$. The tournament runs in two rounds. In the first round A_1 plays against A_2 (the first pair), and A_3 plays against A_4 (the second pair), while in the second round the first round winners play against each other. Return games are not played. There's a time limit for each move, but the depth limits of the algorithms are such that none of the algorithms ever reaches that time limit, which renders the time limit irrelevant. **Based on the information available, which algorithm is expected to win this chess turnament?**

A A_1 or A_3 B A_2 or A_4 C A_3 or A_4 D A_2 or A_3