TECHNISCHE UNIVERSITÄT DRESDEN

Computational Logic ∴ Group

Dr. Hannes Strass

M.A. Jonas Karge

Algorithmic Game Theory

Summer Term 2024

Exercises 4

06–08/05/2024

Problem 1.
Consider a $2 \times n$ zero-sum game where the best response to every pure strategy is unique. Suppose that the top row and leftmost column define a pure-strategy equilibrium of the game.

1. Show that this equilibrium can be found by iterated elimination of strictly dominated strategies.

   Hint: Consider first the case $n = 2$.

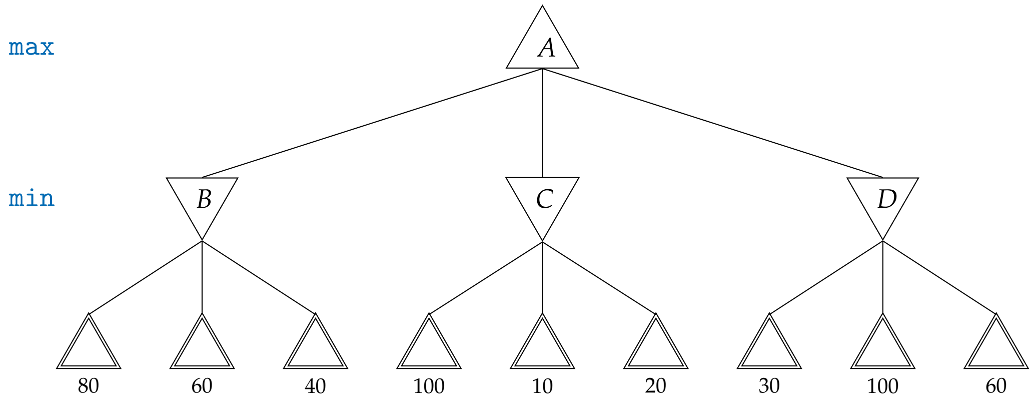
2. Does the claim in (a) still hold if some pure strategy has more than one best response?

Problem 2.
This task exercises the basic concepts of game playing, using Tic-Tac-Toe as an example. We define X_n as the number of rows, columns, or diagonals with exactly n Xs and no Os. Similarly, O_n is the number of rows, columns, or diagonals with just n Os. The utility function assigns $+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as

$$Eval(s) = 3 \cdot X_2(s) + X_1(s) - (3 \cdot O_2(s) + O_1(s))$$

- Approximately how many possible games of Tic-Tac-Toe are there?
- Show the whole game tree starting from an empty board down to depth 2 (i.e., one X and one O on the board), taking symmetry into account.
- Mark on your tree the evaluations of all the positions at depth 2.
- Using the (heuristic) minimax algorithm, mark on your tree the backed-up values for the positions at depth 1 and 0, and use those values to choose the best starting move.
- Circle the nodes at depth 2 that would not be evaluated if alpha-beta pruning were applied, assuming the nodes are generated in the optimal order for alpha-beta pruning.

Problem 3.
Consider the following 2-player zero-sum game displaying the payoffs for the max player.



Do the following:

- Determine for each decision node of a player the best move by application of the minimax algorithm and indicate the minimax value of the game.
- Circle all nodes that would not be explored if the Alpha-Beta algorithm is applied to this game tree.

Problem 4.
Describe how the minimax and alpha-beta algorithms change for two-player, **non-zero-sum** games in which each player has their own utility function. You may assume that each player knows the other's utility function. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha-beta?