Prof. Dr. Goran Glavaš,
M.Sc. Fabian David Schmidt
M.Sc. Benedikt Ebing
Lecture Chair XII for Natural Language Processing, Universität Würzburg

8. Exercise for "Algorithmen, KI & Data Science 1"

1 Adversarial State Space Search

1. Define the tic-tac-toe game as adversarial state space search problem giving the (a) initial state, (b) successor function, (c) terminal function, and (d) payoff function.
2. How might the actual utility of player MAX change compared to its expected utility ((a) increase, (b) decrease, or (c) stay the same), if MAX follows the optimal strategy (MINMAX-method), but MIN does not (e.g., MIN chooses randomly). Give an example game tree or argue why an option is not possible.
3. Does the number of pruned nodes in alpha-beta pruning depend on the ordering of explored successor states? Explain.
4. How could you improve the ordering/selection of successor nodes such that alpha-beta pruning becomes more efficient (i.e., prunes more nodes)?
5. Implement the $ticTacToe$ class in the given $.ipynb$. The class represents a tic-tac-toe game. Given a state s it should return the best possible next move. The state s is represented as a 3×3 matrix, where an empty field is represented by 0, an "X" is represented by 1, and "O" is represented by -1. The two players are "MAX" (using "Xs" or $1s$) and "MIN" (using "O" or $-1s$). Implement the following methods:
   a) Player function $player(s)$: Given a state s return whose turn is next (either "MAX" or "MIN"). Assume that from the initial state MAX always makes the first move.
   b) Successor function $succ(s)$: Given a state s return the set of possible successor states.
   c) Winner function $winner(s)$: Given a state s return $False$ if no one has won

yet or the name of the player (either "MAX" or "MIN") otherwise.

d) Terminal function $terminal(s)$: Given a state s return $True$ if the state is terminal and $False$ otherwise.

e) Utility function $utility(s)$: Given a state s return 1 if "MAX" wins, -1 if "MIN" wins or 0 otherwise

f) Max value function $max\_val(s)$: Given a state s return the maximum of all minimum values (pseudocode in lecture 16, slide 12)

g) Max value function $min\_val(s)$: Given a state s return the minimum of all maximum values (pseudocode in lecture 16, slide 12)

h) Minmax function $minmax(s)$: Given a state s return $False$ if the game ended without winner, return the winner if there is one or return the best possible next state.