Course **Artificial Intelligence Techniques (IN4010)**
Part *Adverarial Search: Game Playing*

Contact Lecturer Dr. Nils Bulling
Teaching Assistant Chris Rozemuller
Course e-mail ai@ii.tudelft.nl

Exercise Sheet **Tutorial 2: Game Playing**
Release date 21 September 2016
Tutorial date 22 September 2016

Tutorial 2: Adversarial Search and Game Playing

This sheet is about adversarial search. Some of these exercises (marked C) will be discussed and presented during the tutorial class. The remaining exercises are for self-study. We very much encourage you to work on some of these exercises at home.

Exercise 1 (Finite and Infinite Game Trees, C)**.** Describe four board games, two of which with a finite, and two with an infinite game tree. In which category falls chess, with and without the "fifty-move rule"[1]? Justify your answers.

Exercise 2 (MiniMax Algorithm)**.** The Minimax algorithm assumes that the opponent plays optimally. Suppose the opponent *does not play optimally*. Show the following.

1. For non-optimally playing opponents the Minimax value is at least as good as the computed Minimax value if played against an optimal opponent.
2. For non-optimally playing opponents, the Minimax value may not be optimal. Construct a *counterexample*.

Exercise 3 (Alpha-Beta Minimax in Non-Zero Sum Games)**.** Consider a two-player, turn-based, **non-zero-sum** game. In such games each player can have an arbitrary utility function (if a terminal node is reached the utility values of both players can be accessed by each of the players). Is it possible for any node to be pruned by Alpha-Beta Minimax (i.e. if we apply the algorithm on such game trees is it still assumed that an optimal solution is found)?

Exercise 4 (Heuristic Search and Resource Limitations, C)**.** Suppose we want to write a computer program playing the 8-puzzle board game (cf. Figure 1). Decisions must be made within 4 seconds where the computer can explore up to 16 nodes per second. We want to use the Heuristic Minimax algorithm to find the optimal action within the given resource bounds. The evaluation function should be based on the total number of misplaced tiles wrt. the goal state. For example, in Figure 1 the evaluation function should return a value of 6 (the less the better).

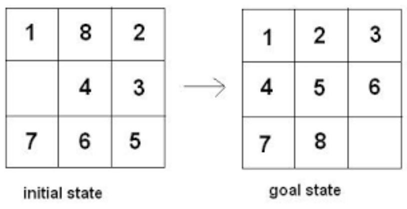

Figure 1: Two state of an 8-puzzle. Left, the initial state and, right, the goal state. The player can make a move by sliding a tile adjecent to the open space into the open space to reach a new configuration. The game ends when the goal state has been reached.

(a) What is the **maximum** and the **average** branching factor for an 8-puzzle game?

(b) How many plies can we search the game tree considering the maximal branching factor and resource constraints? (In your computation use exact values, no approximations.)

[1]fifty-move rule: if no player makes a capture, or moves a pawn for 50 moves the game ends in a draw

(c) Given the initial situation shown below, draw the search tree for 2-plies and calculate the Heuristic Minimax value with respect to this initial situation.

	2	3
4	5	6
7	8	1

(d) Is it a good evaluation function? Will it find a solution under resource constraints?

Exercise 5 (Heuristic Minimax)**.** This exercise is about the board game Tic-Tac-Toe, played on a 3x3 grid.

(a) Given a game tree, we also refer to a branch in the game tree as a *game* (thus, a game captures a possible play of Tic-Tac-Toe). Give an upper bound on how many possible games of Tic-Tac-Toe are there?

(b) Draw the game tree starting from an empty board down to depth 2 (i.e., 2-plies). Take symmetry into account; that is, do not draw any branches in the game that are symmetric, but indicate the number of (left out) symmetric states.

(c) Define an appropriate *evaluation function* for Tic-Tac-Toe. Give arguments why it is a good one and compute the utility of the states at depth 2 from part (b).

(d) Based on the utility of the evaluation function of (c) use the Heuristic Minimax algorithm to compute the Heuristic Minimax value of each state of the game tree. Also, give the corresponding optimal actions.

Exercise 6 (Stochastic 8-puzzle)**.** Consider the problem of solving a variant of the 8-puzzle game (again cf. Figure 1). There are two players and two 8-puzzle games. Suppose we make the problem adversarial as follows: The two players take turns consecutively. A coin is flipped to determine the puzzle on which to make perform an action in that turn. The winner is the first to solve one of the puzzles.

(a) Give an informal proof that someone will eventually win this game if both players always make optimal actions.

(b) Does one of the players have a strategy that guarantees winning? Give arguments.

Exercise 7 (Stochastic Games)**.** Consider carefully the interplay of chance events and incomplete information in each of the following games: Monopoly, Scrabble, or Texas hold'em poker. For which is the Expectiminimax algorithm appropriate?

Exercise 8 (Partially Observable Games, C)**.** A quizmaster hides a prize behind one of three doors. Afterwards, you can also choose one of the three doors. Next the quizmaster reveals one of the doors without the prize which you did not choose. You are allowed to move from your currently selected door to the other closed door, or stick to your initial selection.

(a) What is your optimal strategy? Should you change or not?

(b) Describe the problem if you use the "averaging over clairvoyance" method to compute the "optimal" action