UNIVERSITÄT HEIDELBERG
Zukunft. Seit 1386.

IWR
Interdisciplinary Center for Scientific Computing

Artificial Intelligence
Wintersemester 17/18
Prof. Björn Ommer
ommer@uni-heidelberg.de

Exercise sheet 5: Game Playing

Due on *01/12/2017*, 2pm.

Question 1: Tic Tac Toe (4P)

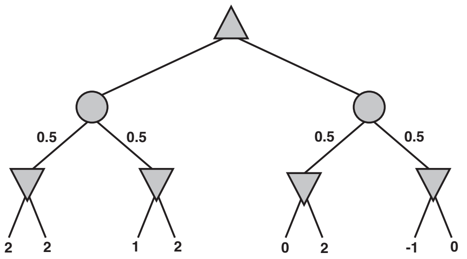
In this exercise we consider the well-known game Tic Tac Toe. If you are not familiar with the rules, please see `https://en.wikipedia.org/wiki/Tic-tac-toe`.

a) Starting from an empty board describe how the minimax algorithm can be applied to determine a strategy. What are suitable utility values? How deep does the tree get at most? How many nodes are processed? For the case that you do not want to process the full tree provide a suiltable evaluation function

b) Implement the minimax algorithm you described before in Python. The algorithm should run to the maximal depth.

Question 2: Pruning (2P)

In this exercise we consider pruning in games with chance nodes. First read the chapter "Games of Chance" in the attached tutorial. The figure below shows the complete game tree for a trivial game. Assume that the leaf nodes are to be evaluated in left-to-right order, and that before a leaf node is evaluated, we know nothing about its value—the range of possible values is $-\infty$ to ∞.



a) Mark the value of all internal nodes in the figure, and indicate the best move at the root.

b) Given the values of the first six leaves, do we need to evaluate the seventh and eighth leaves? Given the values of the first seven leaves, do we need to evaluate the eighth leaf? Explain your answers.

c) Suppose the leaf node values are known to lie between -2 and 2 inclusive. After the first two leaves are evaluated, what is the value range for the left-hand chance node?

d) Identify all the leaves that need not be evaluated under the assumption in (c).

Question 3: Four Dices Straight (4P)

Imaging the following game: We have four dices to roll. The goal is to get a straight (1-4, 2-5 or 3-6). We roll in three rounds. In each round we can choose the number of dices to roll again.
Let assume that we already made a roll (first round) and got

$$1, 1, 3, 4.$$

So we now can choose up to four dices to make a second roll and then again up to four dices for the third round.

a) Define an approach which calculates the best strategy in terms of the highest probability of obtaining a straight, i.e. for a choice s the $p(s)$ provides the change that this choice leads to a straight (ration between straights and all possible outcomes after this choice).

b) Implement this approach in Python, where the probabilities for each possible choice in the second round should be given.

Note: Submit exactly one ZIP file and one PDF file via Moodle before the deadline. The ZIP file should contain your executable code. Make sure that it runs on different operating systems and use relative paths. Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. The PDF file should contain your written code, all figures, explanations and answers to questions. Make sure that plots have informative axis labels, legends and captions.