ICS 271
Fall 2018
Instructor : Kalev Kask
Homework Assignment 3
due Saturday, October 27

1. (25 points) Consider a Tic-Tac-Toe game. Let X_n be the number of rows, columns or diagonals containing exactly n X's and no O's. Similarly, let O_n be the number of rows, columns or diagonals containing exactly n O's and no X's. We propose a utility function which assigns +1 to any position with $X_3 = 1$ (i.e. winning position) and assigns −1 to any position with $O_3 = 1$ (i.e. losing position). The linear heuristic evaluation function we suggest is

$$(4X_2 + X_1) - (4O_2 + O_1).$$

   (a) How many states (i.e. board positions) are there in a Tic-Tac-Toe game. Note that there are symmetric board positions.

   (b) What is the (maximum) depth of the complete game tree (that starts from empty board and has terminal nodes at the bottom)? Does the complete game tree contain all the board positions you counted in (a)?

   (c) Start from a board where there is an X is the center and O in the top left corner and it is X's turn to move. Show the game tree down to depth 2 where there are two X's and two O's on the board.

   (d) Mark on your tree the heuristic evaluation function values of all (leaf) nodes at level 2. Thereafter, mark on your tree the min-max values of all remaining nodes.

   (e) Apply alpha-beta search on your tree, and mark the pruned subtrees when traversing from left to right, from right to left, and in an **optimal order** (resulting in most pruning).

2. (5 points) Assume you are playing a five-piece endgame in chess. Many modern chess programs contain built-in game trees for all endgames with limited number of pieces (such as five-piece endgames). Assume leaves of this tree are labeled by −1 if the position is a losing position, +1 if it is a winning position and 0 if it is a tie. Is algorithm alpha-beta pruning guaranteed find a winning strategy whenever it exists? Why?

3. (10 points) Consider the game tree in Figure 1 in which the terminal node scores are all from player MAX's point of view.

   (a) What move should the first player choose?

   (b) What nodes do not need to be examined using the alpha-beta algorithm – assuming that nodes are examined in left-to-right order?

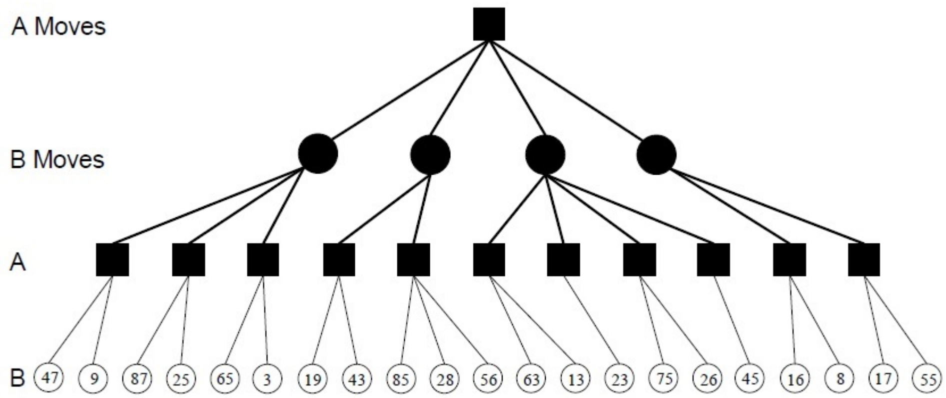

Figure 1: A Game tree

4. (10 points) Prove that with a positive linear transformation of leaf values (e.g., transforming a value x to $6ax + b$, where $a > 0$), the choice of move remains unchanged in a game tree.

5. (10 points) (5.19 in RN) Consider the following procedure for choosing moves in games with chance nodes:

   - Generate some dice-roll sequences (say, 50) down to a suitable depth (say, 8).
   - With known dice rolls, the game tree becomes deterministic. For each dice-roll sequence, solve the resulting deterministic game tree using alpha–beta.
   - Use the results to estimate the value of each move and to choose the best.

   Will this procedure work well? Why (or why not)?