

ICS 271

Fall 2015

Instructor : Kalev Kask

Homework Assignment 3

due Tuesday, 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 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 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)? Does it contain additional board positions?
 - (c) Start from a board where there is an X in 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 evaluation function values for all nodes at level 2 (i.e. the value of the utility function at each node). Thereafter, mark on your tree the min-max values.
 - (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*.
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 static scores (in parentheses at the leaf nodes) are all from the first player's point of view. Assume that the first player is the maximizing player.
 - (a) What move should the first player choose?

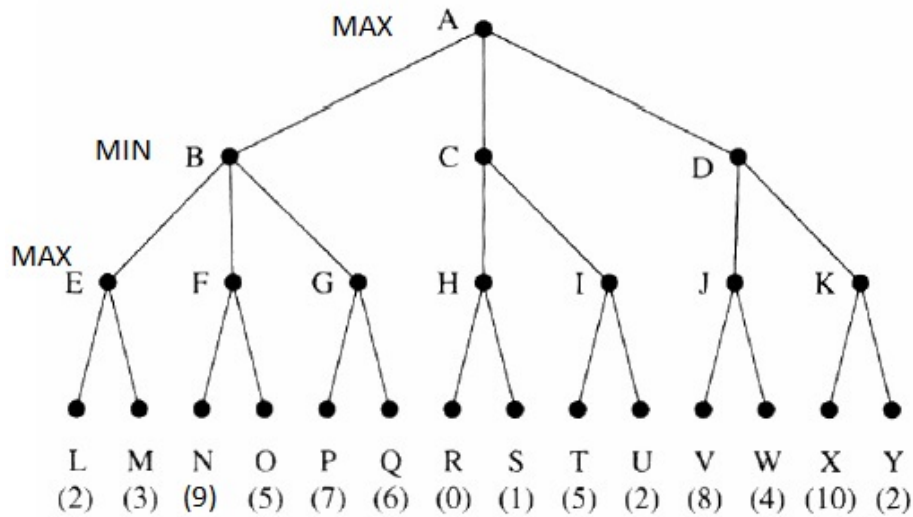


Figure 1: A Game tree

- (b) What nodes would not need to be examined using the alpha-beta algorithm—assuming that nodes are examined in left-to-right order?
4. (10 points) Prove that with a positive linear transformation of leaf values (e.g., transforming a value x to $2ax + b$, where $a > 0$), the choice of move remains unchanged in a game tree.
5. (10 points) Consider the following procedure for choosing moves in games with chance nodes.
- Lets assume that you are searching a non-deterministic game tree with a dice-roll before each players move, to depth 8. Lets assume the game tree has m chance nodes. Further, you have generated, in advance, 1000 dice-roll sequences of length m (i.e. a total of $1000m$ dice rolls).
 - Assume you will execute the following procedure 1000 times,
- (a) Search the game tree using alpha/beta search; at each chance node, use a precomputed dice-roll outcome. Note that with known dice rolls, the game tree is deterministic.

Take the average over all 1000 executions to determine the best move. Will this procedure work well in detemining the best move? Why, or why not?