

ARTIFICIAL INTELLIGENCE COMP813

AUCKLAND UNIVERSITY OF TECHNOLOGY M3 EXERCISES

ADVERSARIAL SEARCH AND GAMES

Question 1. Tic-Tac-Toe.

This problem exercises the basic concepts of game playing, using tic-tac-toe (naughts and crosses) as an example, so that you have a better understanding on building the Deep Blue beating the best human player.

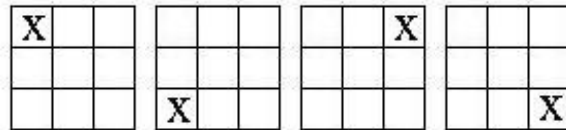
We define X_n as the number of rows, columns or diagonals with exactly n X's and no O's. Similarly, O_n is the number of rows, columns or diagonals with just n O's. The utility function assigns +10 to any position with $X_3 \geq 1$ and -10 to any position with $O_3 \geq 1$. All other terminal positions have utility 0. For the nonterminal positions, we use a linear evaluation function defined as

$$Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$$

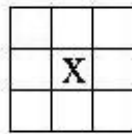
The game start with Xplayer.

- (a) Approximately how many possible games of tic-tac-toe are there? One game play is a sequence of moves from initial state till a terminal state.
- (b) Show the whole game tree starting from an empty board down to depth 2 (i.e. one X and one O on the board), taking **symmetry** into account.

By symmetry, we mean that if you turn the whole board around (without moving the relative position of the pieces), you'll get the same configuration. For example, the following 4 states are the same under symmetry.



They are different from the following state:



- (c) Mark on your tree the evaluations of all the positions at depth 2.
- (d) Using the minimax algorithm, mark on your tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.
- (e) Circle the nodes at depth 2 that would not be evaluated if alpha-beta pruning were applied, assuming the nodes are generated in the optimal order for alpha-beta pruning.

Question 2. We played the Take-away game during the lecture.

- (a) Represent the game in a game tree. Estimate how many decision nodes are there for each players and how many strategies are there for each players. Show your working.
- (b) Does a player have a winning strategy (a way to enforce a win no matter what the other player does)? Show your working.
- (c) Analyse the game using the Minimax algorithm.
- (d) Implement the minimax algorithm to test your results in (a) and (b).