1 Games and Adversarial search (25/10/2018)

1.1 Objectives

At the end of this repetition you should be able to:

- Define formally the search problem associated to a game (IPATTU[1])
- Define and apply the Minimax algorithm
- Define and apply $\alpha - \beta$ version of Minimax algorithms
- Define H-Minimax, Expectiminimax, Monte-Carlo Tree Search

1.2 Exercises

a Tic Tac Toe (≈ 30 min)

"*Tic-tac-toe (also known as noughts and crosses or Xs and Os) is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.*"[2]
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 $+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For nonterminal positions, we use a linear evaluation function defined as $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

1. Define the search problem associated with the tic tac toe game.
2. Approximately how many possible games of tic-tac-toe are there?
3. 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.
4. Mark on your tree the evaluations of all the positions at depth 2.
5. 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.
6. 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.

b Chess and transposition table (≈ 15 min)

Suppose you have a chess program that can evaluate 16 million nodes per second. Decide on a compact representation of a game state for storage in a transposition table.

1. About how many entries can you fit in a 4-gigabyte in-memory table?
2. Will that be enough for the three minutes of search allocated for one move?
3. How many table lookups can you do in the time it would take to do one evaluation? Suppose that you have a 3,2GHz machine and that it takes 20 operations to do one lookup on the transposition table.

c Quizz (≈ 15 min)

1. In a fully observable, turn-taking, zero-sum game between two perfectly rational players, it does not help the first player to know what strategy the second player is using that is, what move the second player will make, given the first player's move.
2. What is a quiescent position?
3. What is encouraged by each term of the sum in the formula: $\frac{Q(n',p)}{N(n')} + c\sqrt{\frac{2\log N(n)}{N(n'}}$

[1]Initial state - Player function - Actions function - Transition model - Terminal test - Utility function
[2]https://en.wikipedia.org/wiki/Tic-tac-toe

4. Which heuristic is correct (see image below)?

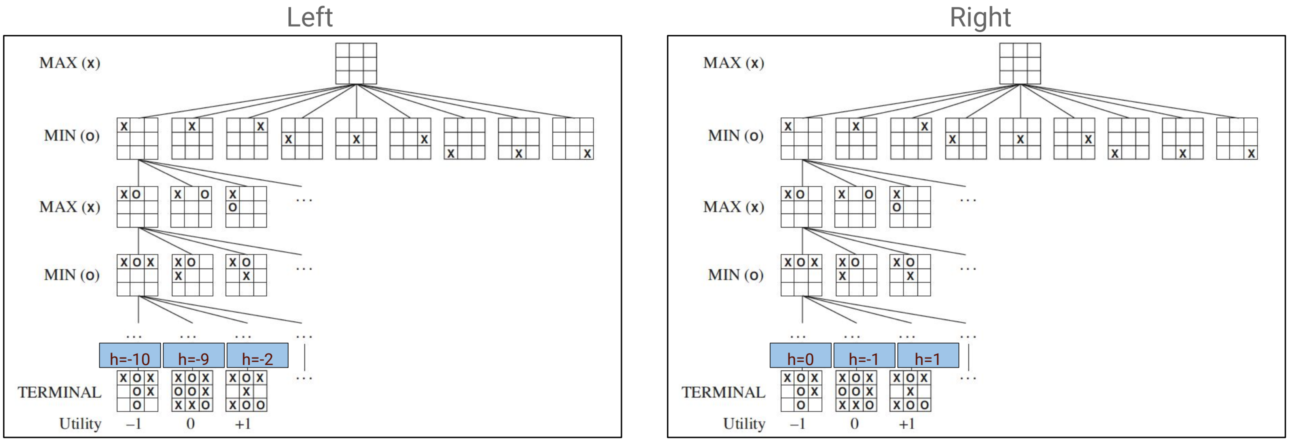

Figure 1: Two possible heuristics