

CS4804 Homework 2

Homework must be submitted electronically following the instructions on the course homepage. Make sure to explain your reasoning or show your derivations. You will lose points for unjustified answers, even if they are correct (and you can earn points for incorrect answers if their justifications are strong).

Written Problems

(Based on 5.10 in R+N) Consider the family of generalized tic-tac-toe games, defined as follows. Each particular game is specified by a set \mathcal{S} of squares and a collection \mathcal{W} of winning positions. Each winning position is a subset of \mathcal{S} . For example, in standard tic-tac-toe, \mathcal{S} is a set of 9 squares and \mathcal{W} is a collection of 8 subsets of \mathcal{S} : the three rows, the three columns, and the two diagonals. In other respects, the game is identical to standard tic-tac-toe. Starting from an empty board, players alternate placing their marks on an empty square. A player who marks every square in a winning position wins the game. It is a tie if all squares are marked and neither player has won.

1. (5 points) Let $N = |\mathcal{S}|$, the number of squares. Give a nontrivial¹, upper bound on the number of nodes in the complete game tree for generalized tic-tac-toe as a function of N . Explain why it is an upper bound. It may help to start by deriving how many leaves there are in this game tree.

You will be graded based on (a) whether your bound is correct, (b) how reasonable your analysis is, and (c) your explanation of why your bound is an upper bound, including discussion of why the number of nodes is actually less than your bound (i.e., what simplifying assumptions you made to make the bound easier to analyze).

2. (5 points) Propose a plausible evaluation function that can be used for any instance of generalized tic-tac-toe. The function may depend on \mathcal{S} and \mathcal{W} .

You will be graded on (a) whether your evaluation function gives a better score to winning positions than the initial (empty) board, (b) whether your function accounts for how close X is to winning and how close O is to winning, (c) whether your description of your function is clear on how to implement it, and (d) the quality of your justification and explanation for your function.

3. (5 points) Assume that it is possible to generate a new board and check whether it is a winning position in $100N$ machine instructions and assume a 2 gigahertz processor. Ignore memory limitations. Using your estimate in (a), roughly how large a game tree can be completely solved by alpha-beta in a second of CPU time? a minute? an hour?

You will be graded on (a) discussion of assumptions you make to perform this analysis, (b) whether your analysis is correct given your assumptions, (c) discussion of best-case and worst-case behavior, and (d) how clear your analysis and justification are.

¹You must provide justification why your bound isn't overly loose. For example, you would not get credit for reporting a bound of ∞ or (one bazillion) ^{N^{N^N}} .