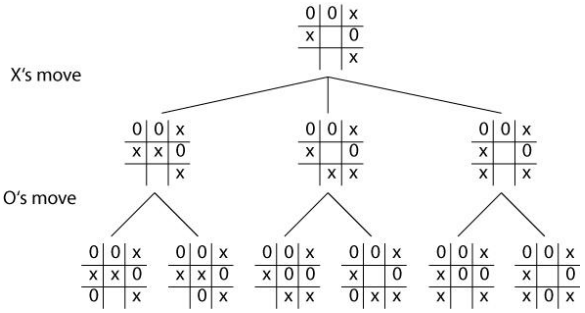
1 Introduction

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn based games such as Tic-Tac-Toe, Mancala, Chess, etc. The minimax algorithm (AI based) is used to determine which moves a computer player makes in games. These kinds of games are called games of perfect information because it is possible to see all possible moves.

2 How the algorithm works

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to get the lowest score possible while minimizer tries to do opposite. we will apply the algorithm on a Tic Tac Toe game,

2.1 Game tree



Representing the game as a game tree allows the computer to evaluate each of its current possible moves by determining whether it will ultimately result in a win or a loss. Game tree in our code is represented by a MiniMax recursive function there is 9! possible probs in the game tree. The code will act like:

- The board state. In this case, where the X's and O's are.
- The current player - the player who will be making the next move.
- The next available moves. For humans, a move involves placing a game token. For the computer, it's a matter of selecting the next game state.
- The game state - the grouping of the three previous concepts.

To sum up, Game Tree is a structure for organizing all possible (legal) game states by the moves which allow you to transition from one game state to the next. This structure is ideal for allowing the computer to evaluate which moves to make because, by traversing the game tree, a computer can easily "foresee" the outcome of a move and thus "decide" whether to take it.

2.2 How can the AI choose the best move based on the game tree

AI should select the winning move. The way we ensure this is to give each move a numerical value based on its board state. We used the following rankings:
Win / O: 1 Lose / X: -1 Draw: 0
It is important is that winning corresponds to the highest ranking, losing to the lowest, and a draw's between the two.
Since the lowest-ranked moves correspond with the worst outcomes and highest-ranked moves correspond with the best outcomes, we should choose the move with the highest value. This is the "max" part of "minimax". the rank of an intermediate Game State where X is the current player should be set to the minimum rank of the available moves. That's what the "mini" in "minimax" refers to.