Tic-tac-toe project (*Documentation*)

1- **Problem formulation:**

- initial state is an empty grid of 3x3.

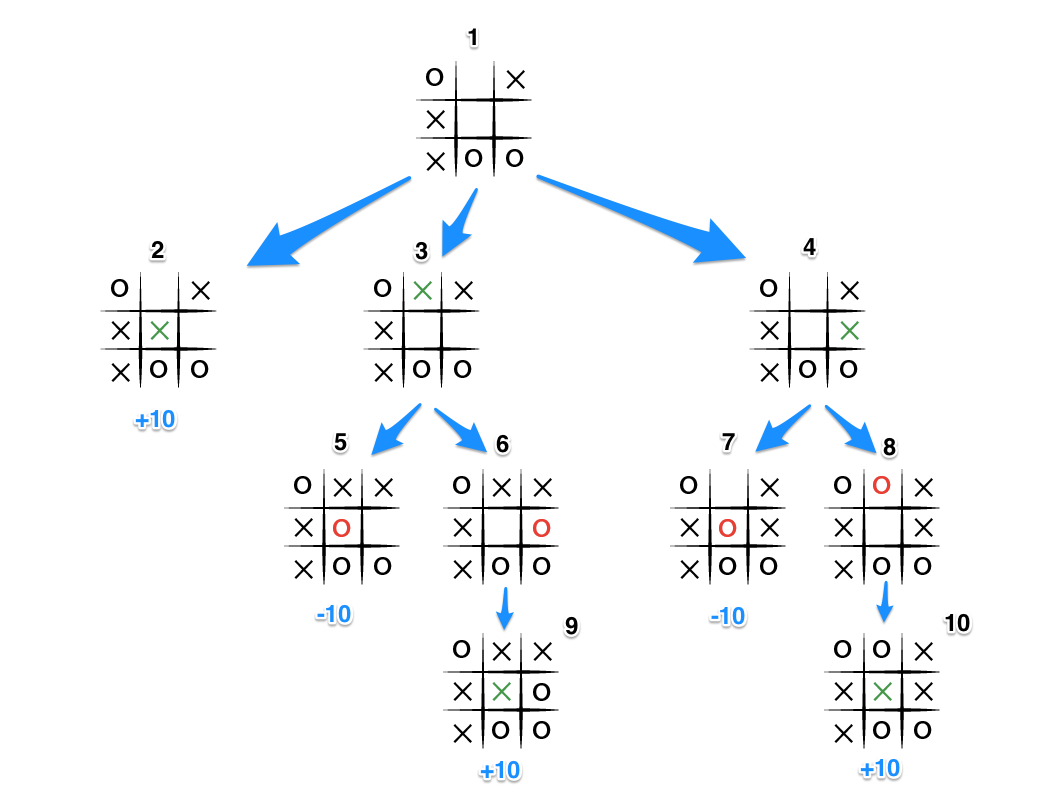
- successor function is the next possible move to best match our goal state to win the game.

- goal test is to win the game by having a row, column or a diagonal match.

- path cost is equivalent to 1 move on the board.

1. **Agent type:** Goal-based agent.
2. **Search tree algorithm** used is **Minimax** search which is a breadth-first-search algorithm, which **processes all the nodes at a level before moving to a node in next level**. It is also a**recursive or backtracking algorithm** which is used in decision-making and game theory. It provides an optimal move for the player assuming that opponent is also playing optimally. Mini-Max algorithm uses recursion to search through the game-tree. Min-Max algorithm is mostly used for game playing in AI.

* And here is a illustration of how Minimax algorithm works in our game Tic-tac-toe.



**As implementing on (X):**

It increases X by 10 when the player plays the final winning move, and it decreases also by 10 when (O) blocks (X) from winning.