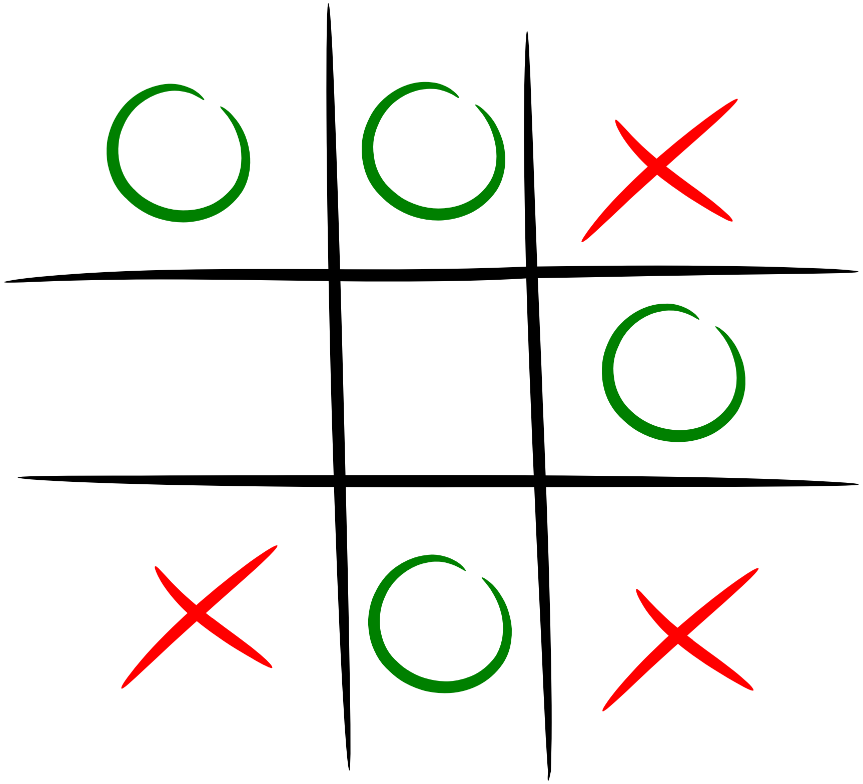
Algorithms for Computer Games

How computer plays board games ?

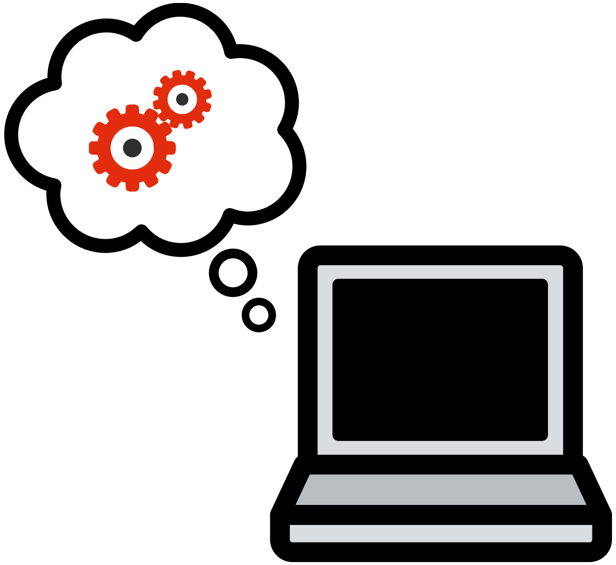
25. Re1

Let's focus on a simple game...

- TicTacToe
  - 2 players (**X** and **O**)
  - alternately put symbols in an empty squares of a 3×3 grid
  - goal : placing three respective marks in a horizontal, vertical, or diagonal row
  - the game is a draw if the grid has been filled without alignments



Let's choose an algorithm to let the computer plays this game...

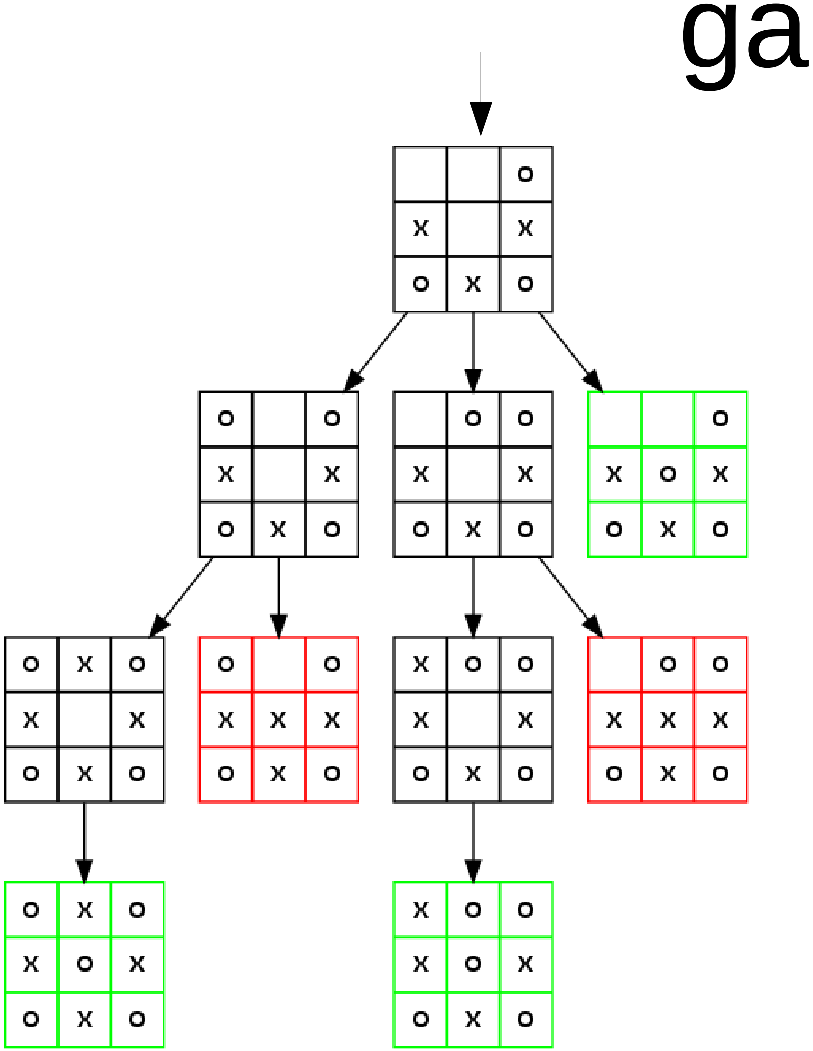


« Minimax »

- easy to understand
- well suited for tictactoe

I'll explain how Minimax works

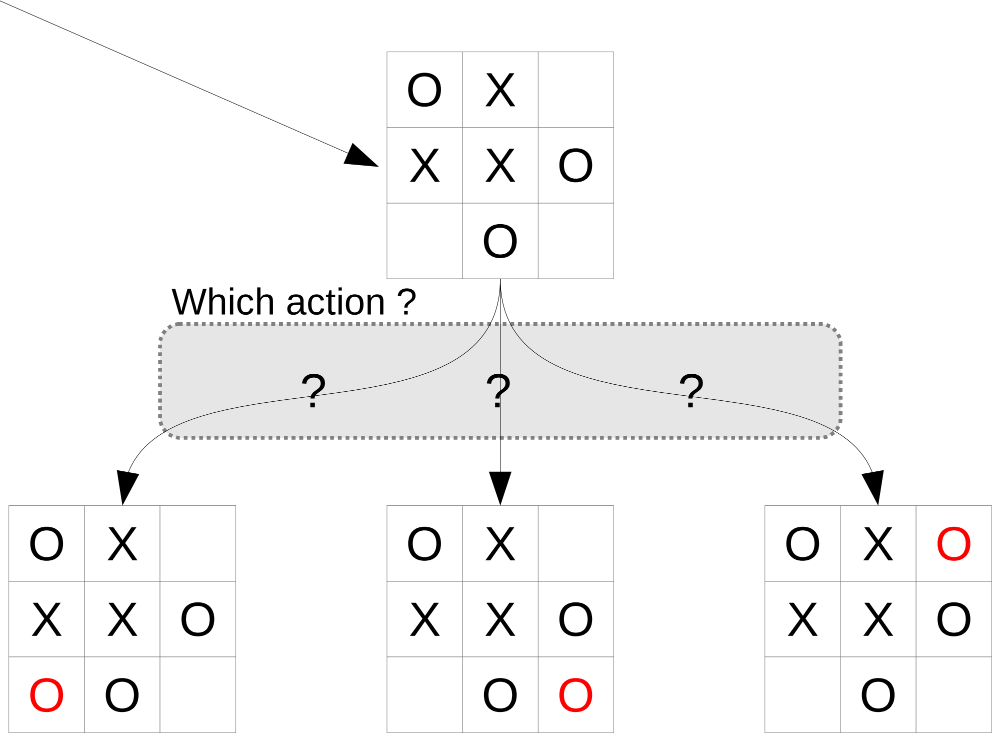
How the computer « see » the game ?



- For the computer, tictactoe is :
  - a set of **states**
  - linked by possible **actions**
  - making a **tree** structure
  - which represents all possible games from a given state

Which action to choose (given a particular state) ?

- Let's say the game is in this state and it's computer's turns
- How it would choose its next move (or action) ?
- The computer will assess the 3 possible next states and choose the action which leads to the best one



O	X	
X	X	O
	O	

O	X	
X	X	O
O	O	

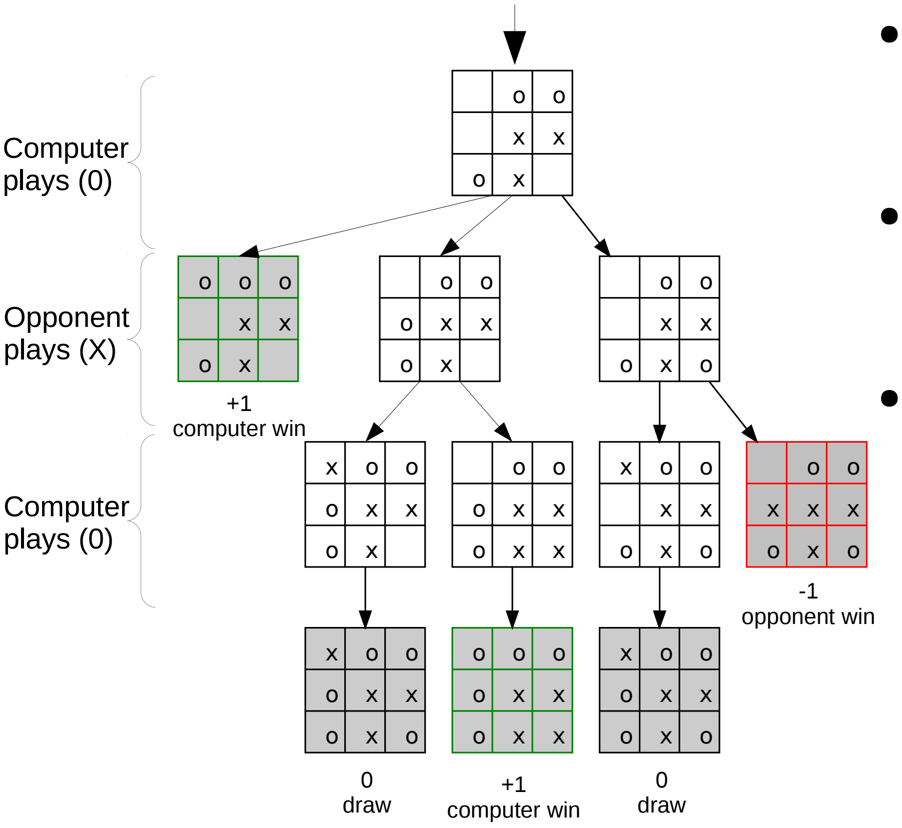
O	X	
X	X	O
	O	O

O	X	O
X	X	O
	O	

How to assess states ?

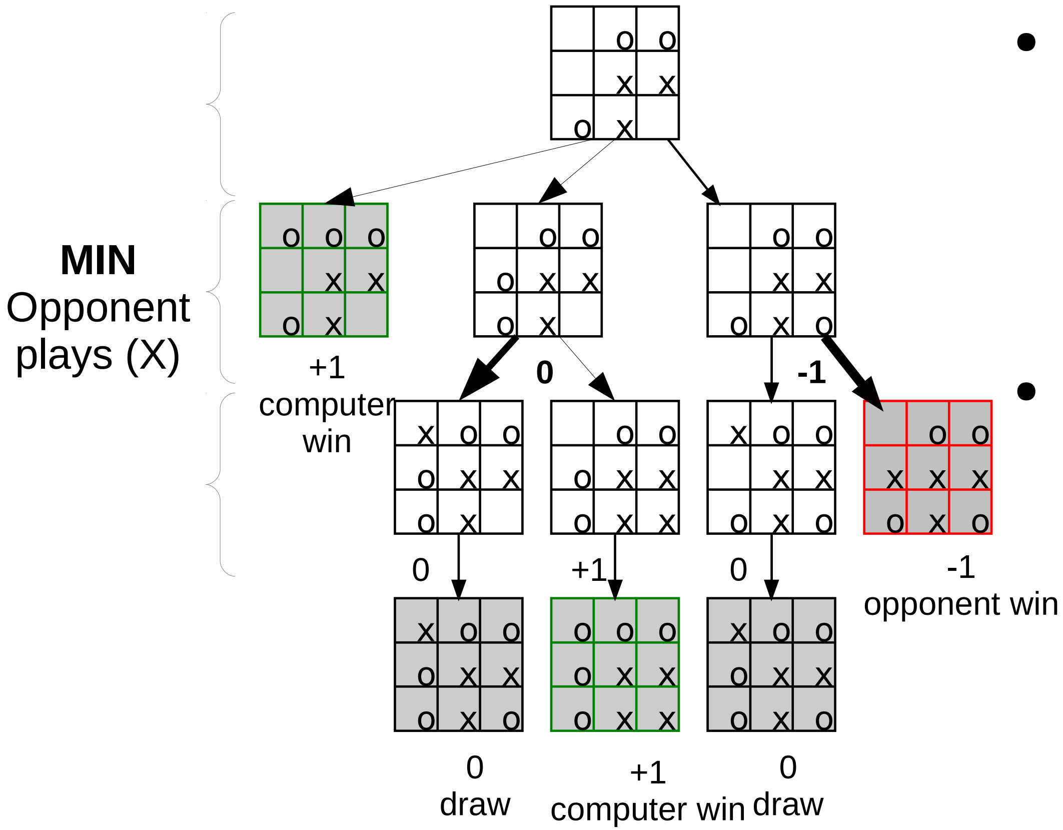
An easy way to do it is to start the evaluation form the end of each possible games

How to assess terminal states ?



- Let's say the game is in this state, it's computer's turns and it plays '**0**'
- In this example, there are 5 terminal states (grey background)
- Let's say terminal states have value
  - +1 if the computer has won
  - -1 if the opponent has won
  - 0 if the game is a draw

And for others states ?



MIN
Opponent plays (X)

+1
computer win

0

-1

0

+1

0

-1
opponent win

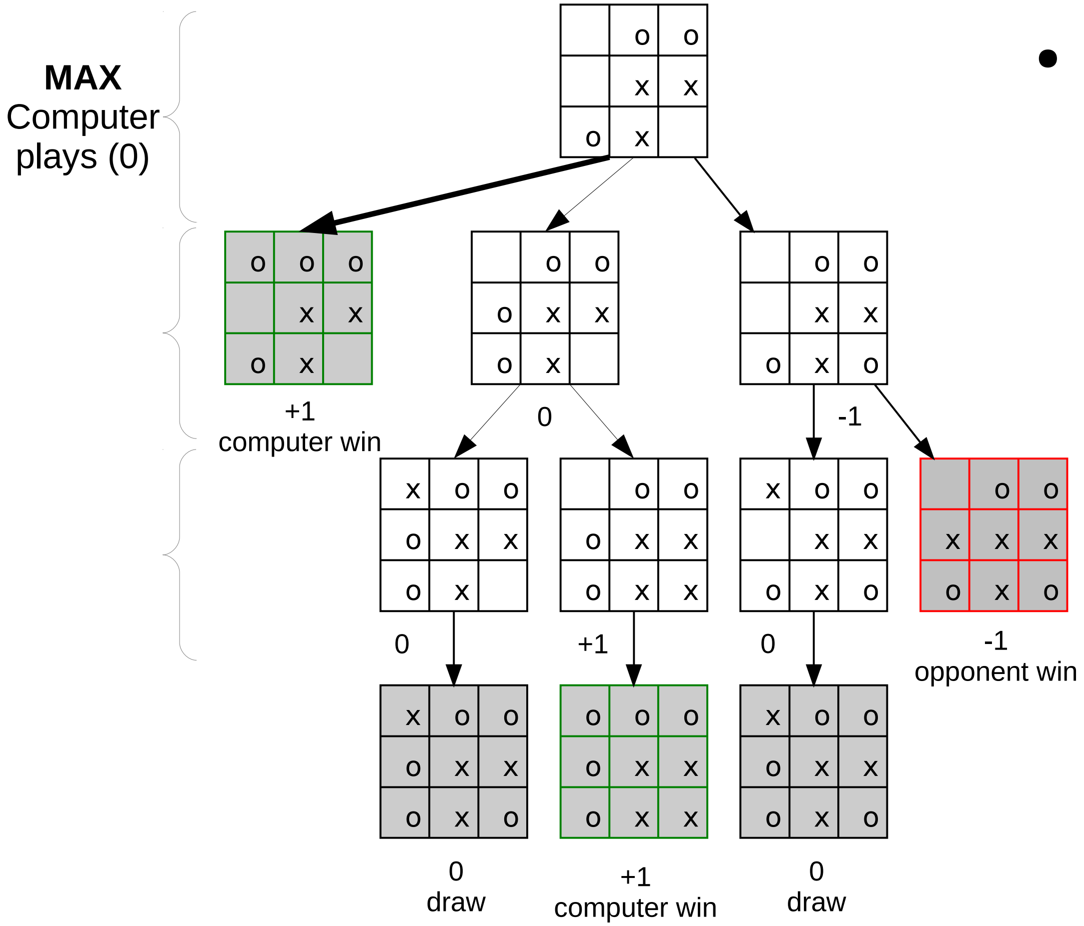
0
draw

+1
computer win

0
draw

- If the opponent plays, we guess he will choose the action which leads to best state for him.
- We considere he will try to :
  - Go to a state « -1 »
  - If not possible, go to a state « 0 »

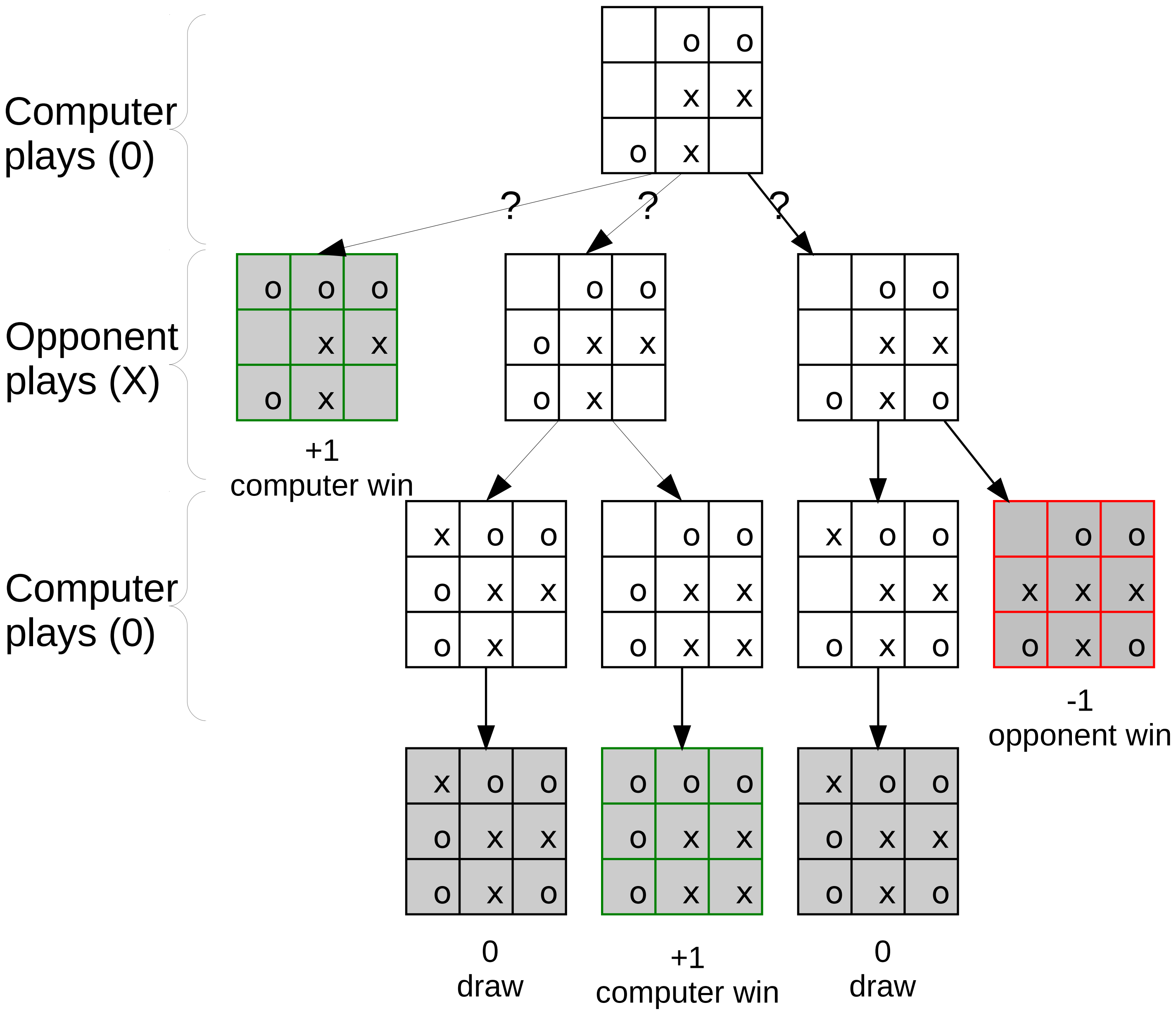
And for others states ?



- If the computer plays, it will choose the action which leads to best state for it :
  - Go to a state « +1 »
  - If not possible, go to a state « 0 »

To sum up

- Each time the computer plays, it has to choose an action from the current state
- To do so, it evaluates all following states from the end of each possible games going back to the current state
- For each evaluation, it choose :
  - the max of the next states if the computer plays
  - the min of the next states if the opponent plays



Computer plays (0)

Opponent plays (X)

Computer plays (0)

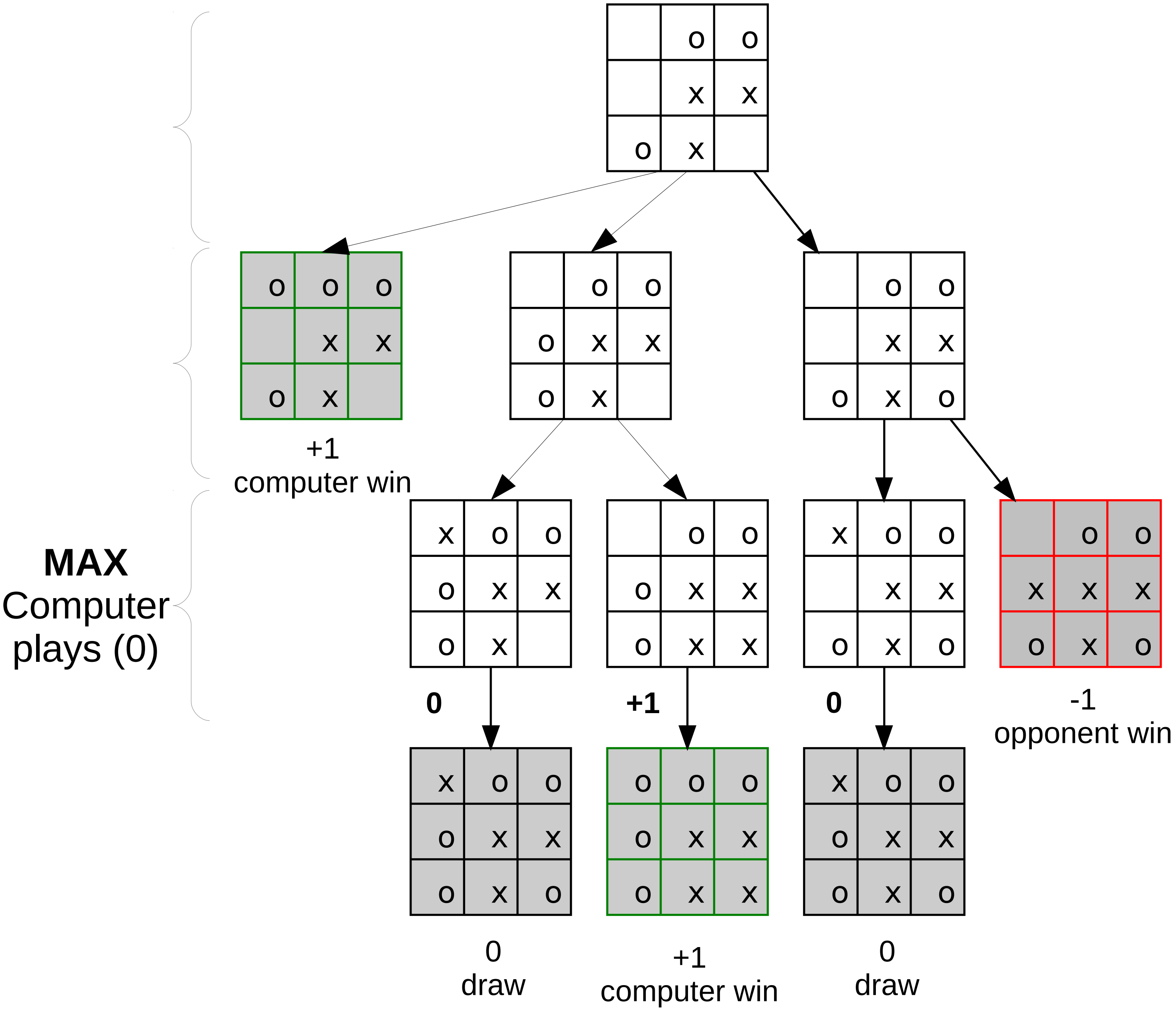
+1 computer win

0 draw

+1 computer win

0 draw

-1 opponent win



MAX
Computer plays (0)

+1
computer win

0

+1

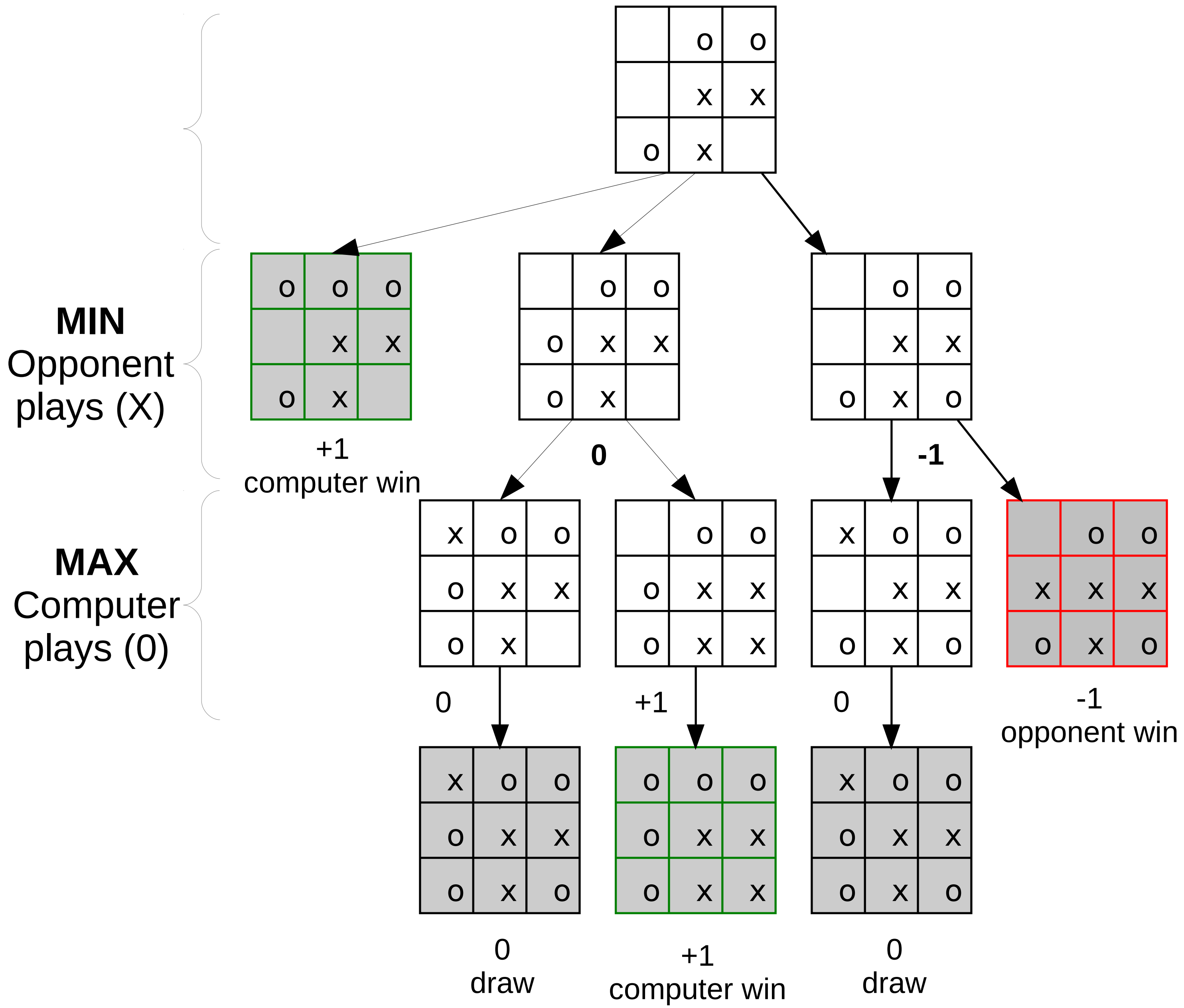
0

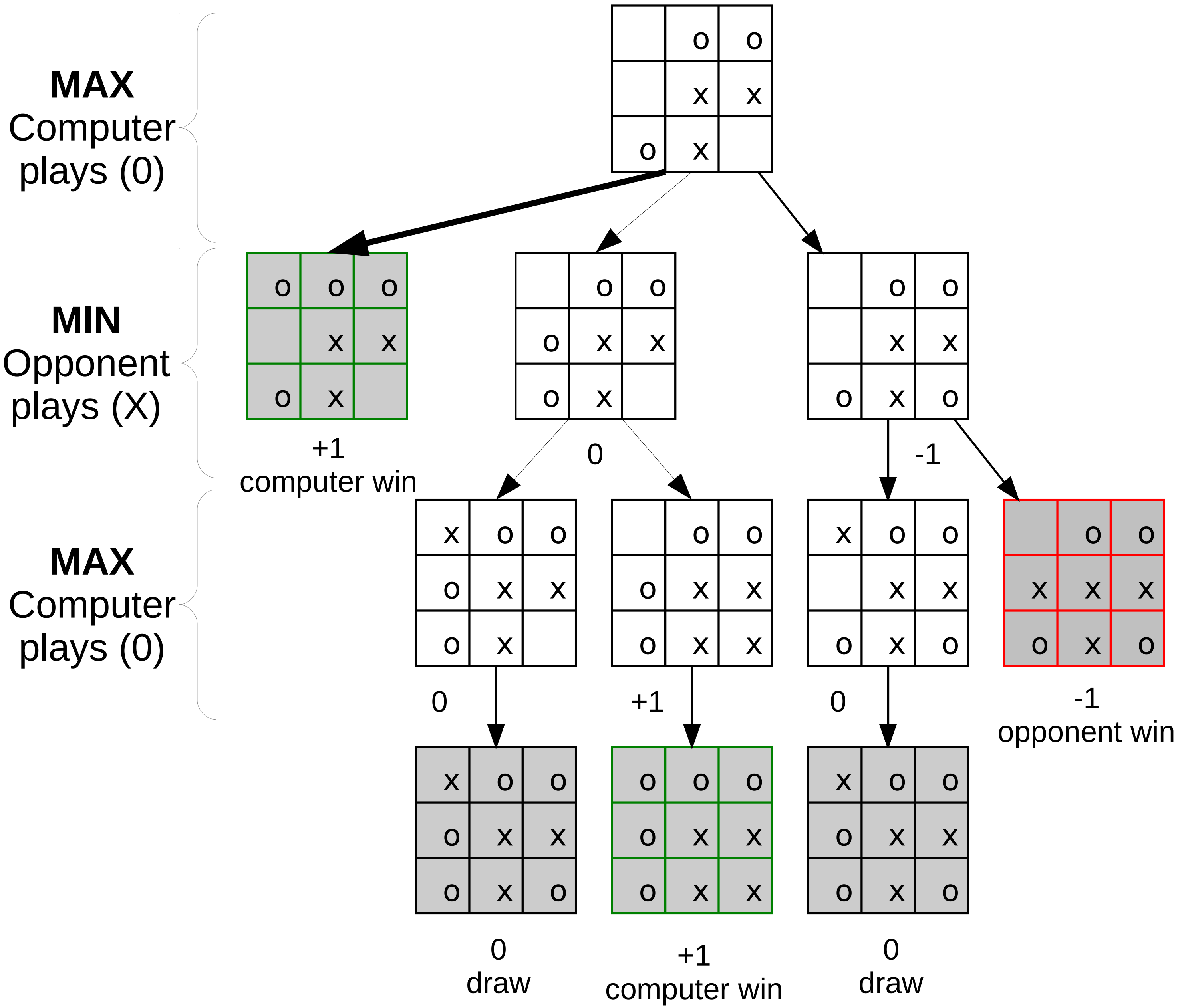
-1
opponent win

0
draw

+1
computer win

0
draw


MIN
Opponent
plays (X)
MAX
Computer
plays (0)
+1
computer win
0
-1
0
+1
0
-1
opponent win
0
draw
+1
computer win
0
draw



MAX
Computer
plays (0)

MIN
Opponent
plays (X)

MAX
Computer
plays (0)

Thank you for your attention

Questions ?