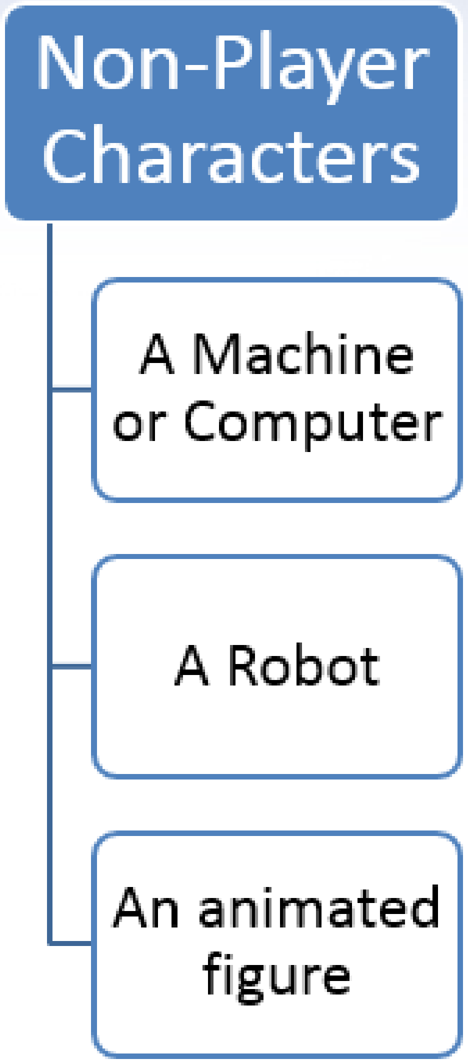
Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Game Playing in Artificial Intelligence

Prof. Bhavisha Suthar

What is Game Playing in AI?

- Game Artificial Intelligence refers to technique used in computer and video games to produce the ***illusion of "intelligence"*** in the behavior of Non-Player Characters(NPCs)



Where is Artificial Intelligence in Game Playing?

- Game must feel natural as Artificial Intelligence does following :
  - -Whether rules are followed or not?
  - -Characters aware of the environment
  - -Path finding (A*)
  - -Decision making
  - -Planning

- Goal: To Obtain solution with unpredictable opponent and time constraints.

Importance of Game Playing in Artificial Intelligence

- Games are fun!
- Game has limited, **well-defined rules**
- They are one of the few domains that allow us to build **agents**.
- Studying games teaches us **how to deal with other agents** trying to foil our plans
- Huge state spaces – Games are highly complex! Usually, there is not enough time to work out the **perfect move**. E.g Chess
- Game playing is considered an **intelligent human activity**.

History

- **Minimax**

  - Developed by John von Neumann in 1928

  - This algorithm is used extensively in game theory

- **Samuel's learning program (1959)**

  - The program learns through the manipulation of the summation of heuristics.

  - If the program wins, it raises high heuristic values and lowers low ones. If it loses, it does the opposite.

- **1960s**

  - Progress and success in Game AI.

  - Creating a successful AI meant coming up with the right rules for it to follow.

- **1970s-1980s**

  - Transition to games as entertainment

  - Game play is based more on skill than on rules

Types of Games

Perfect Information Game:

In which player knows all the possible moves of himself and opponent and their results.

Tic-Tac-Toe, Checkers, Go

Imperfect Information Game:

In which player does not know all the possible moves of the opponent.

Chess, Bridge, Poker

Heuristic Function

(Where we try to choose smarty)

Heuristic Function

- A heuristic is a rule for choosing a branch in a state space search that will most likely lead to a problem solution

→ When used?

- there is no exact solution to a problem, as in medical diagnosis
- there is an exact solution but the computation is prohibitively expensive, as in the game of chess

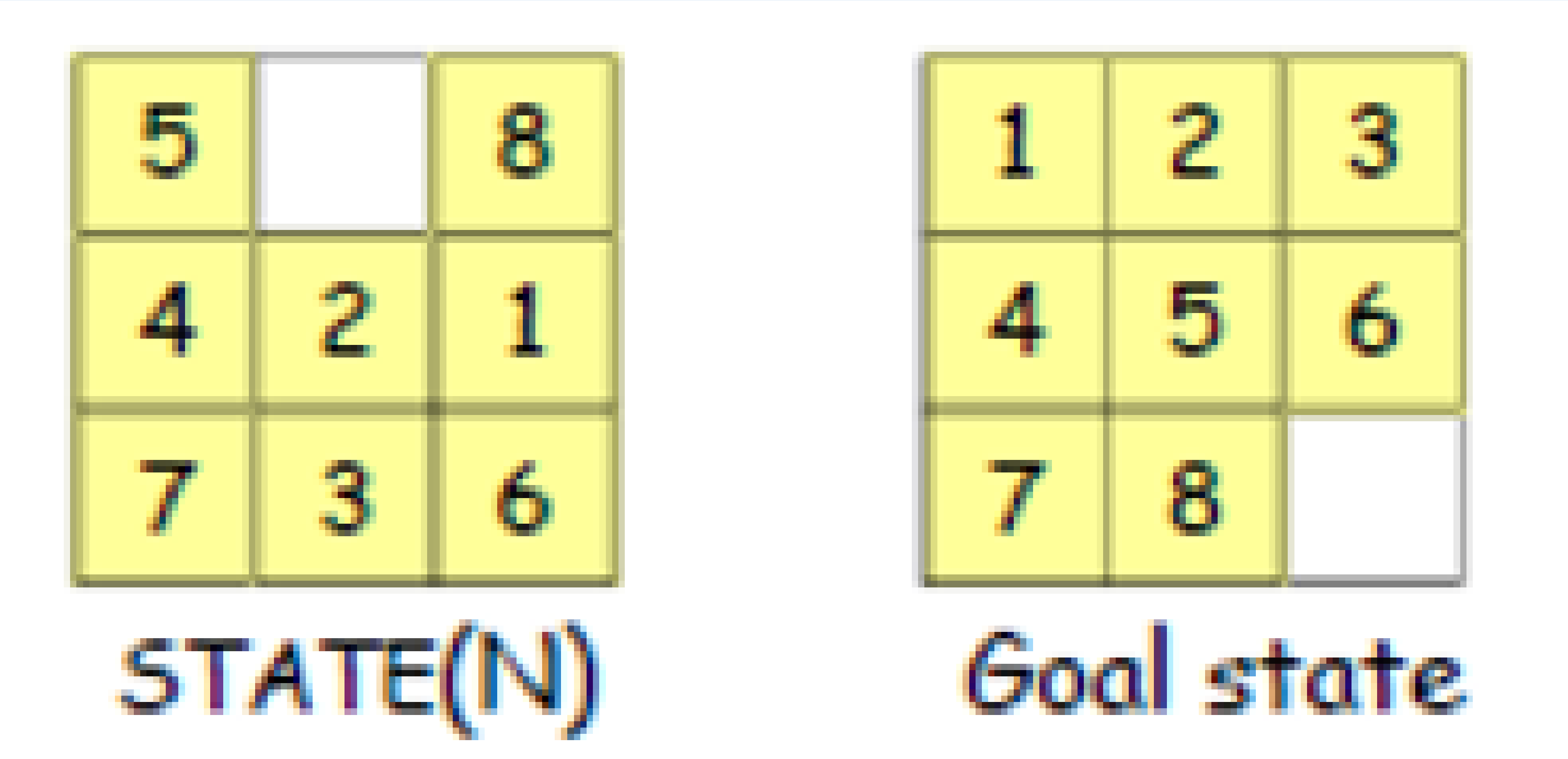
→ Heuristics are fallible

- they may find suboptimal solutions
- they may find no solution at all

- Heuristics are problem dependent and there may be many alternative heuristics for the same problem

For 1 Payer

8-Puzzle Problem



Heuristic function: 8-Puzzle Problem

$h_1(N)$ = number of misplaced numbered tiles = 6

$h_2(N)$ = sum of the (Manhattan) distance of every numbered tile to its goal position

$= 2 + 3 + 0 + 1 + 3 + 0 + 3 + 1 = 13$

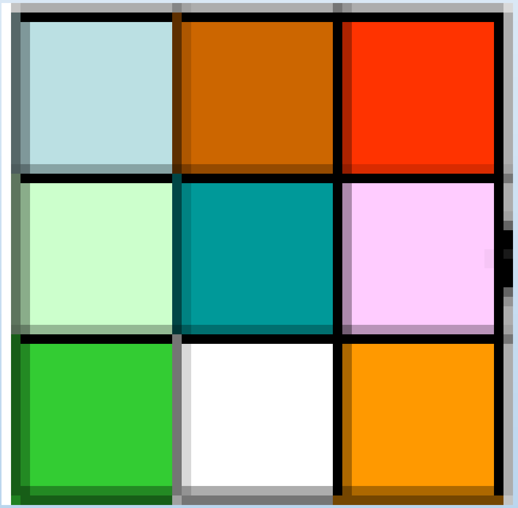
$h_3(N)$ = sum of permutation inversions

$= n_5 + n_8 + n_4 + n_2 + n_1 + n_7 + n_3 + n_6$

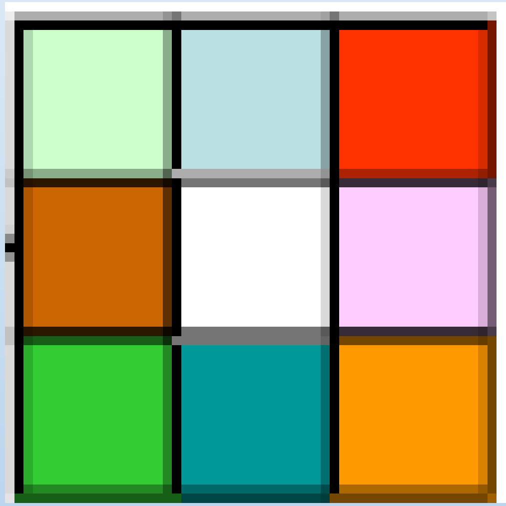
$= 4 + 6 + 3 + 1 + 0 + 2 + 0 + 0$

$= 16$

8-Puzzle Problem

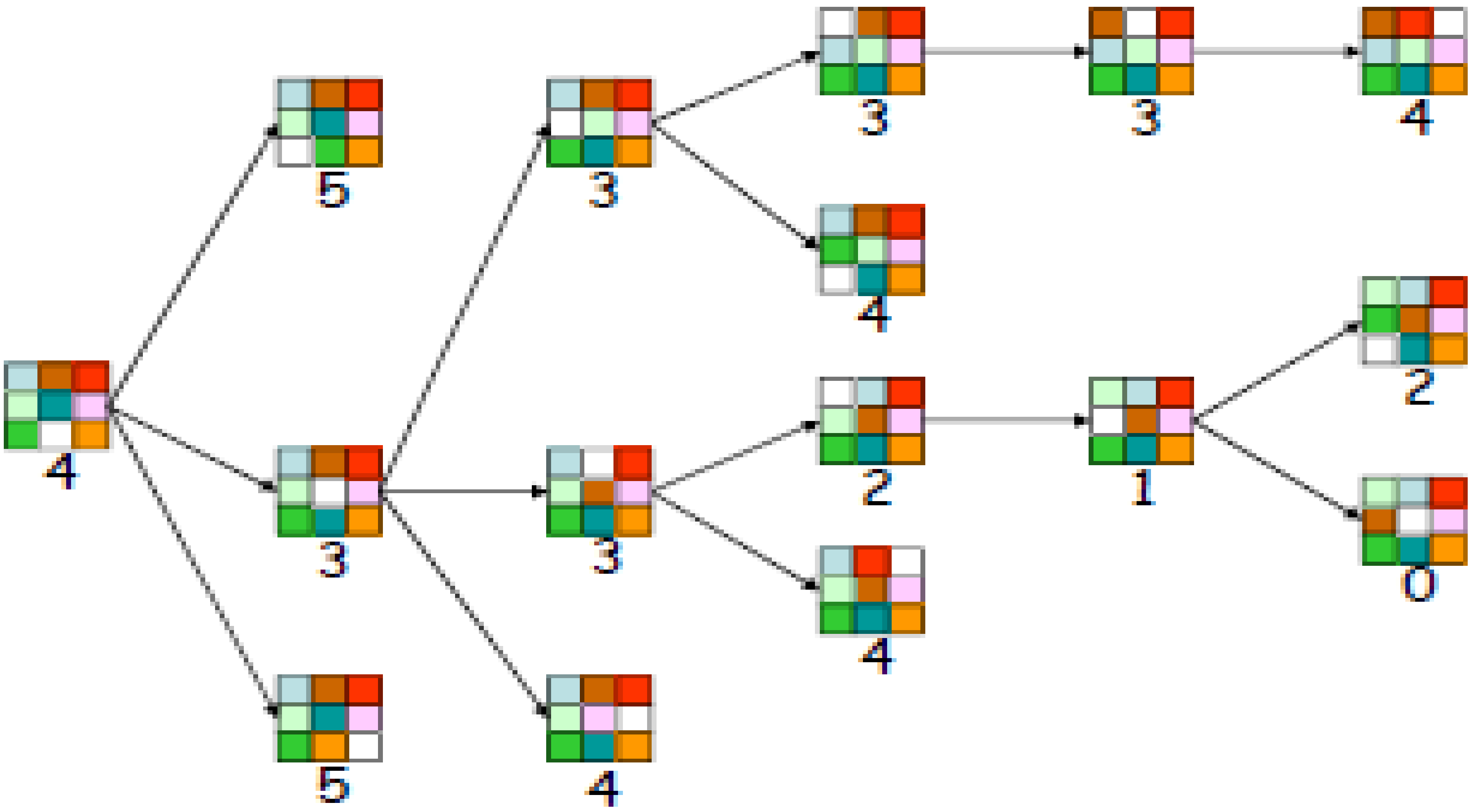


Initial State(N)



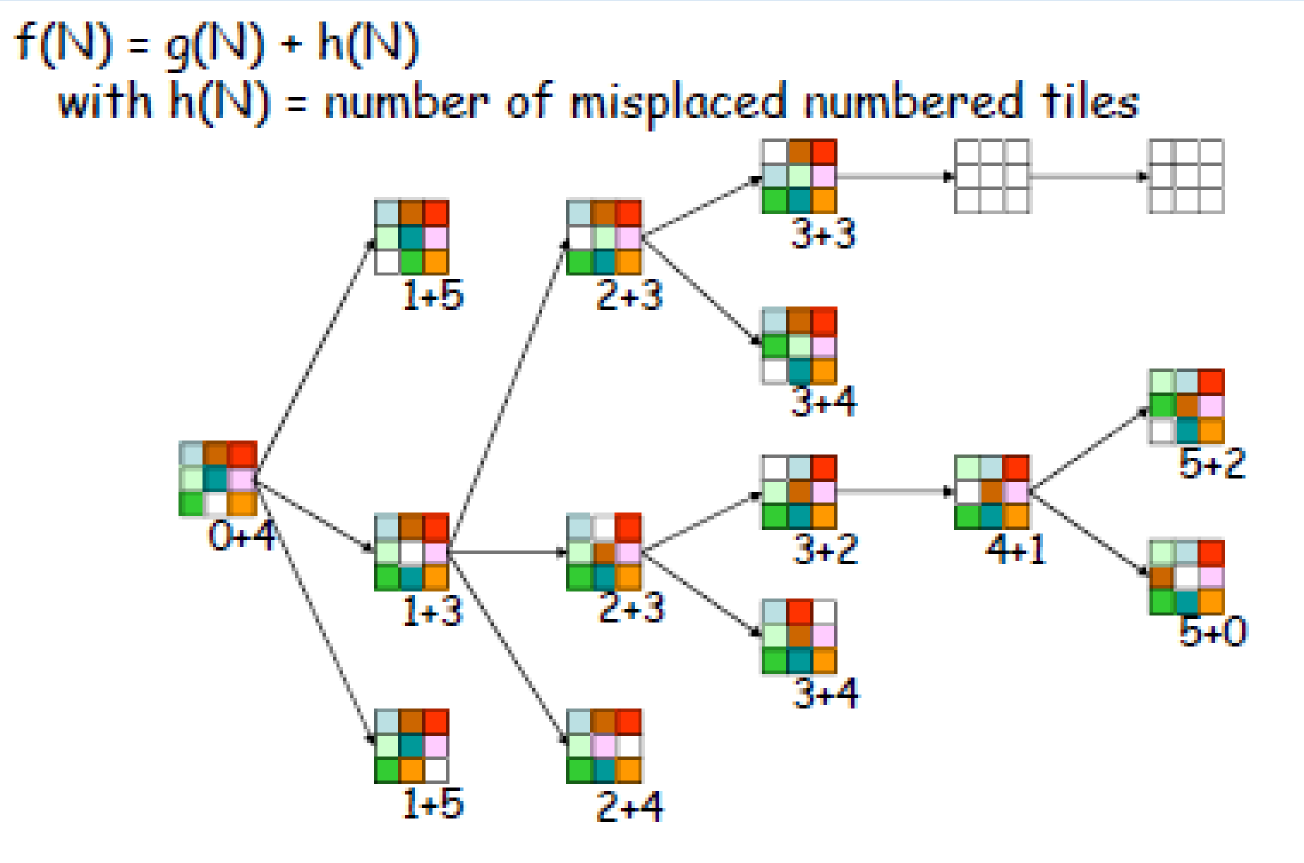
Goal State

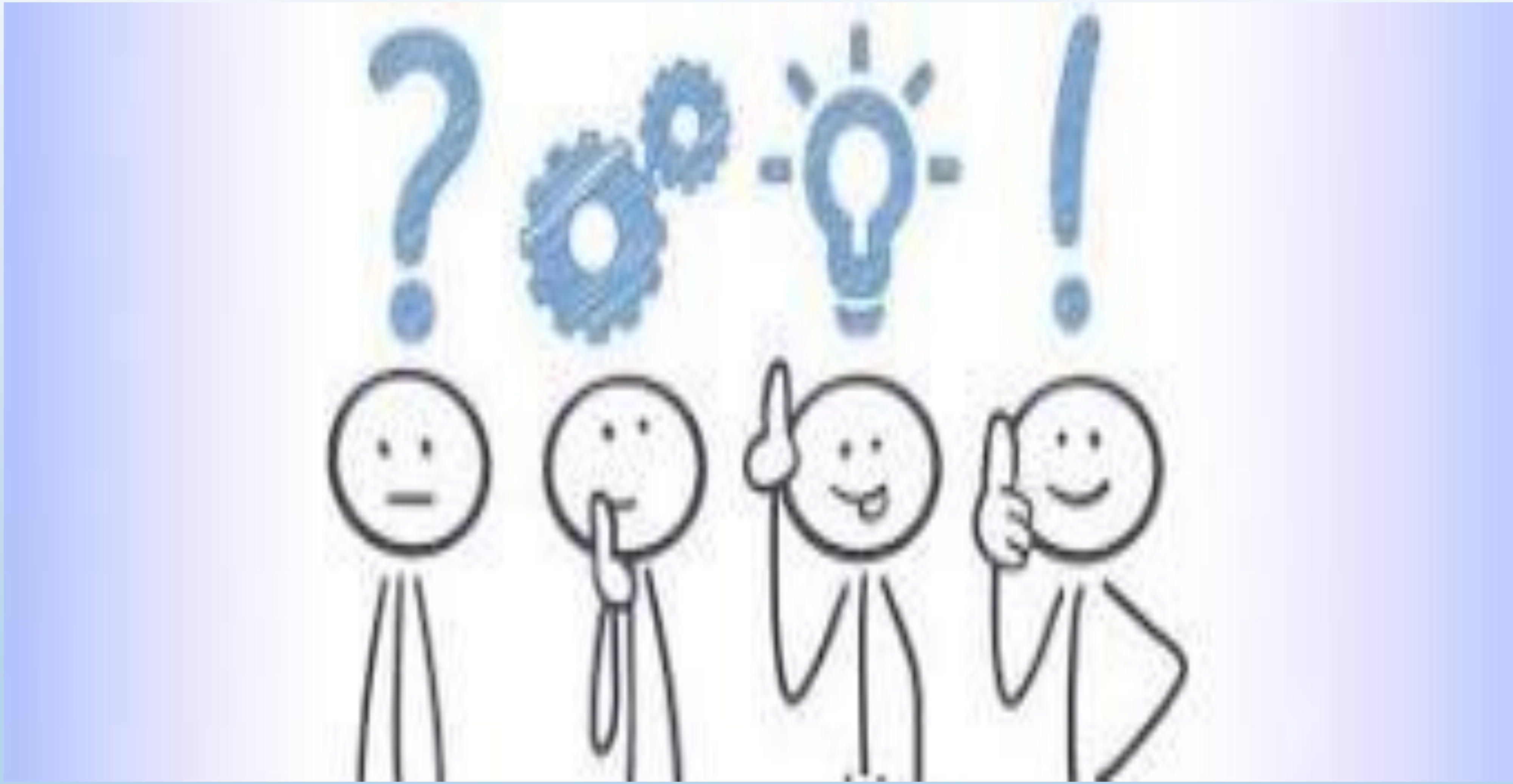
f(N) = h(N) = number of misplaced numbered tiles



The white tile is the empty tile

Solution: 8-Puzzle Problem





8-puzzle?????

For 2 Payers

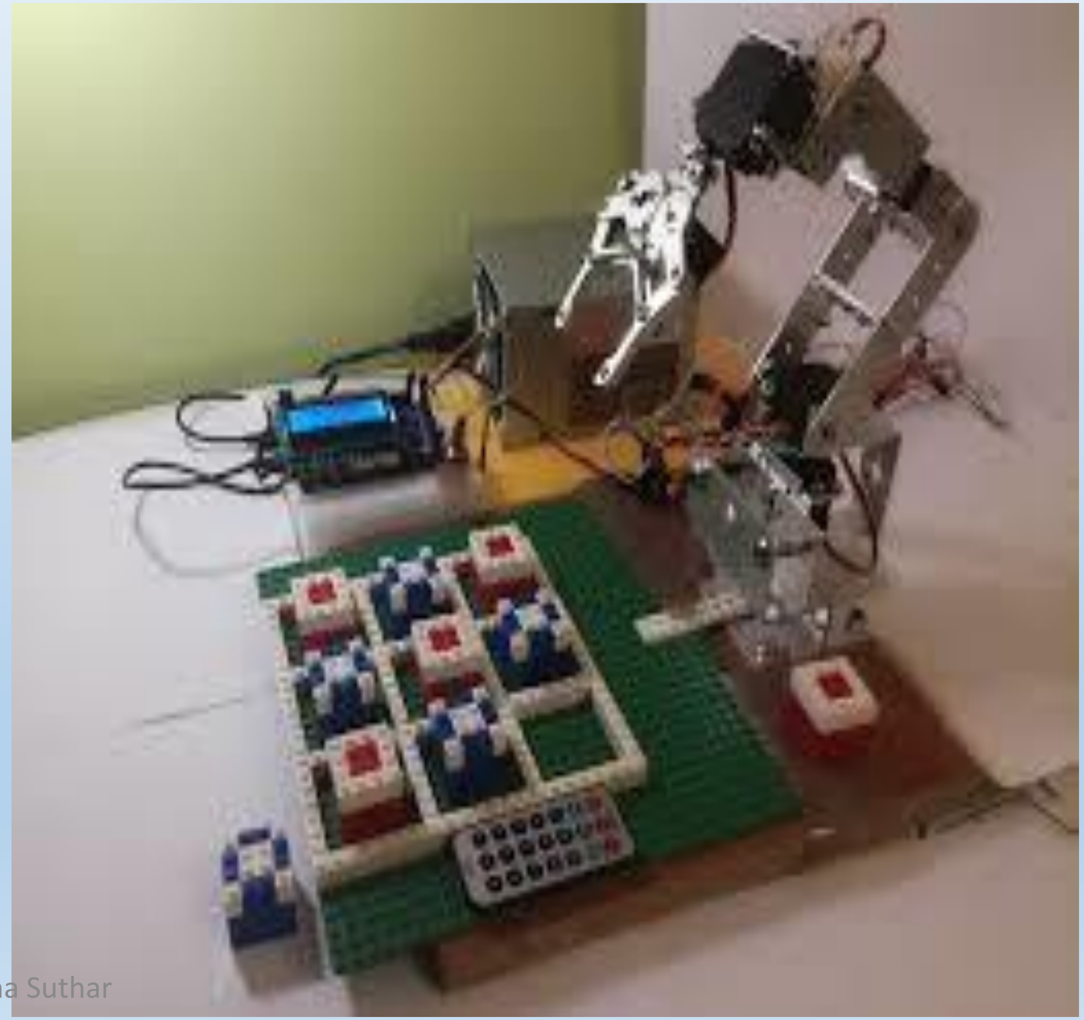
Solving 2-players Games

- **Statement of Game as a Search Problem:**
  - States = board configurations
  - Operators = legal moves. The transition model
  - Initial State = current configuration
  - Goal = winning configuration
  - payoff function (utility)= gives numerical value of outcome of the game

- The primary game theory is Mini-Max algorithm.

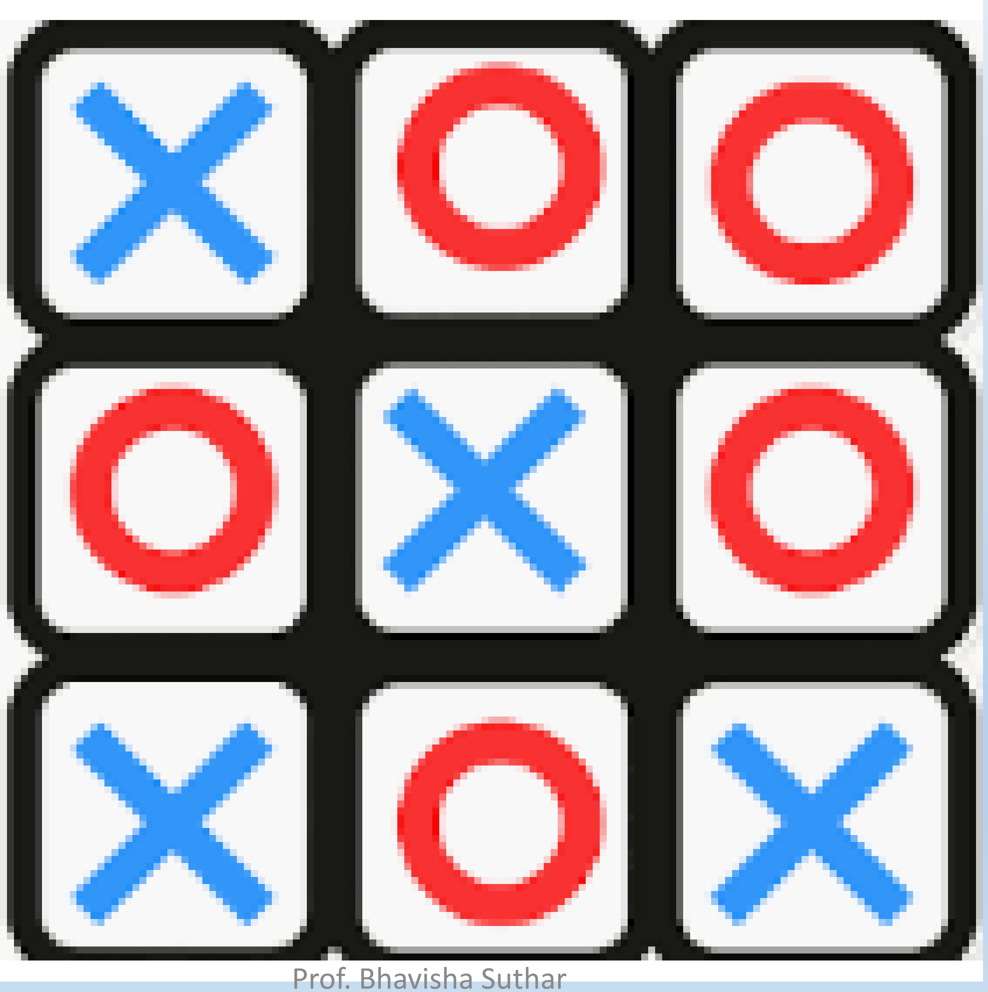
" If a minimax of one player corresponds to a maximin of the other player, then that outcome is the best both player can hope for.

Tic-Tac-Toe Game



Tic-Tac-Toe Game

Crosses



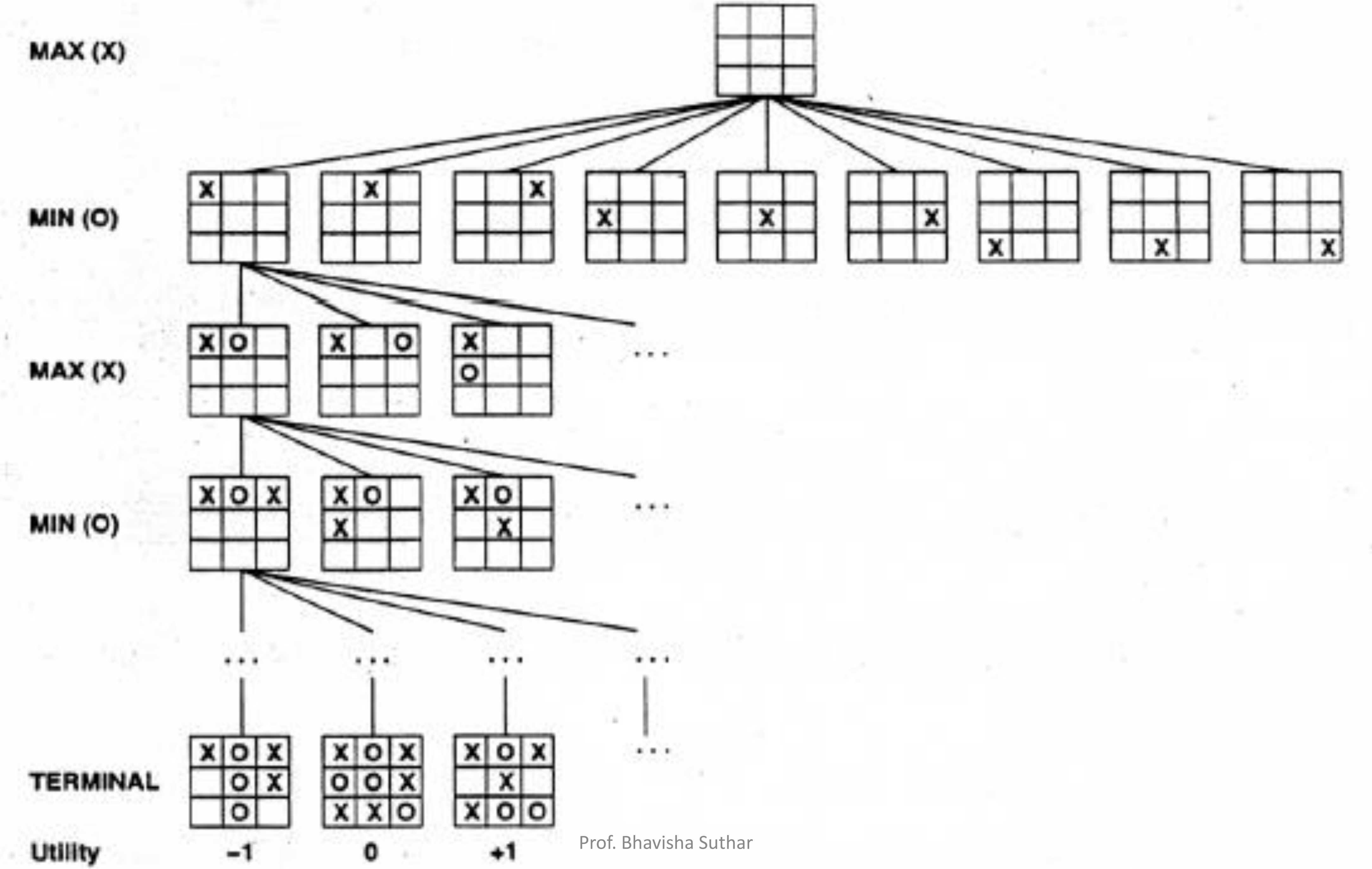
Noughts

Tic-Tac-Toe Game

- **Elements of Mini-Max procedure:**
    - Game tree (search tree)
    - Static evaluation: +ve for win, -ve for lose, o for draw or neutral
    - Backing up the evaluation, level by level, on the basis of opponent's turn.

Tic-Tac-Toe Game

- **General principles of game-playing and search**
  - evaluation functions
  - minimax principle
  - alpha-beta-pruning


MAX (X)
MIN (O)
MAX (X)
MIN (O)
TERMINAL
Utility
-1
0
+1

Heuristic function: Tic-Tac-Toe Game

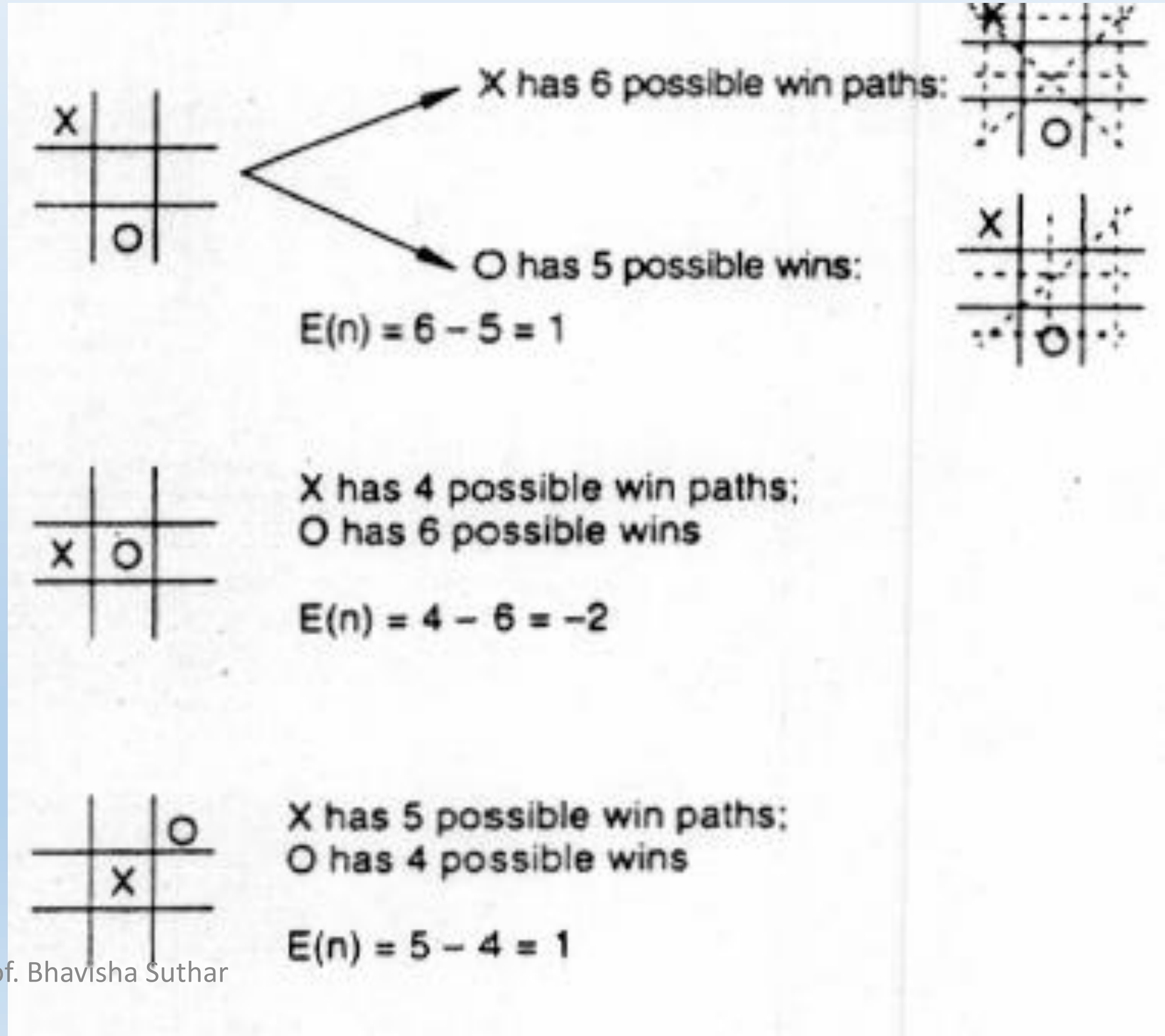
Heuristic is

$$E(n) = M(n) - O(n)$$

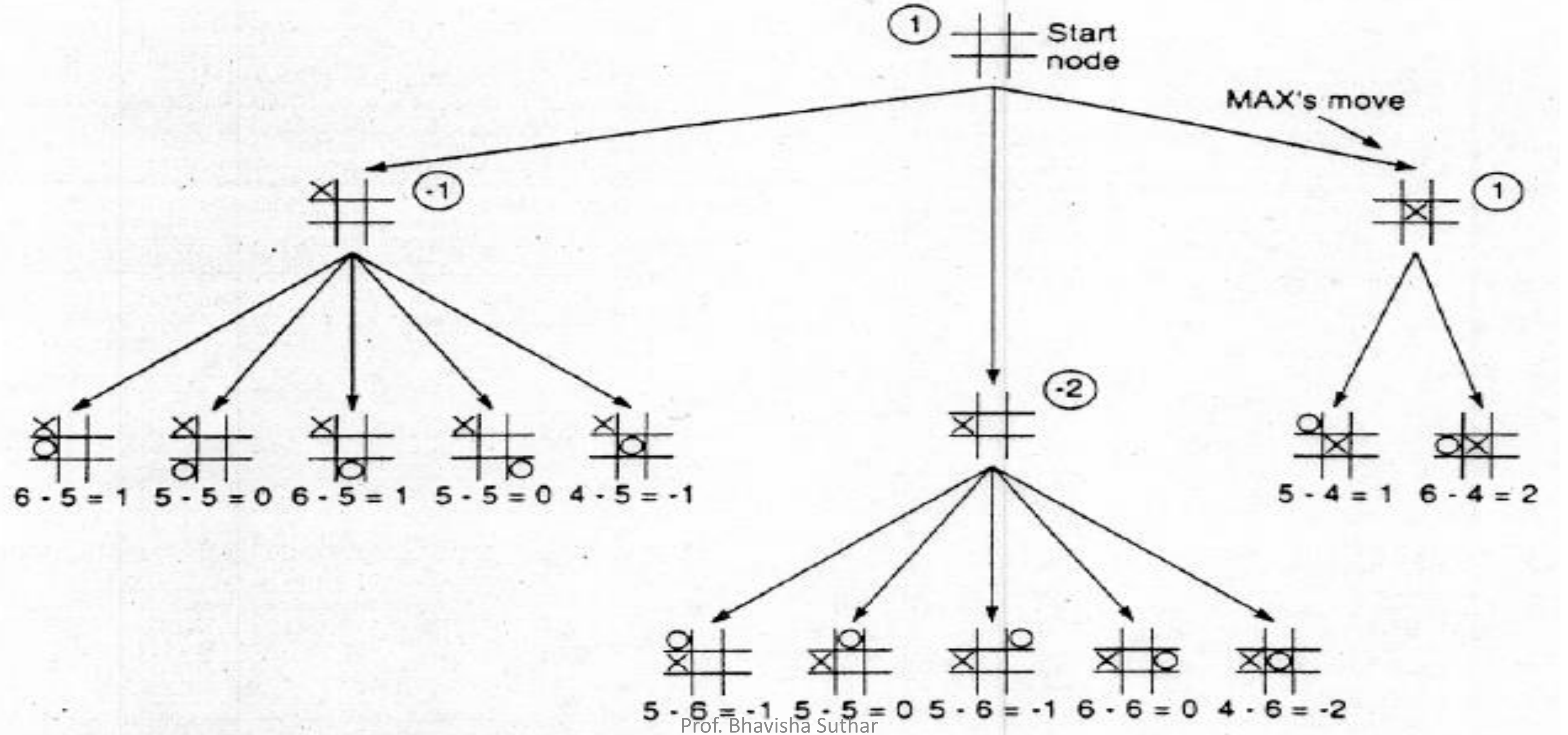
M(n): total no. of my possible winning lines

O(n): total no. of opponent's possible winning lines

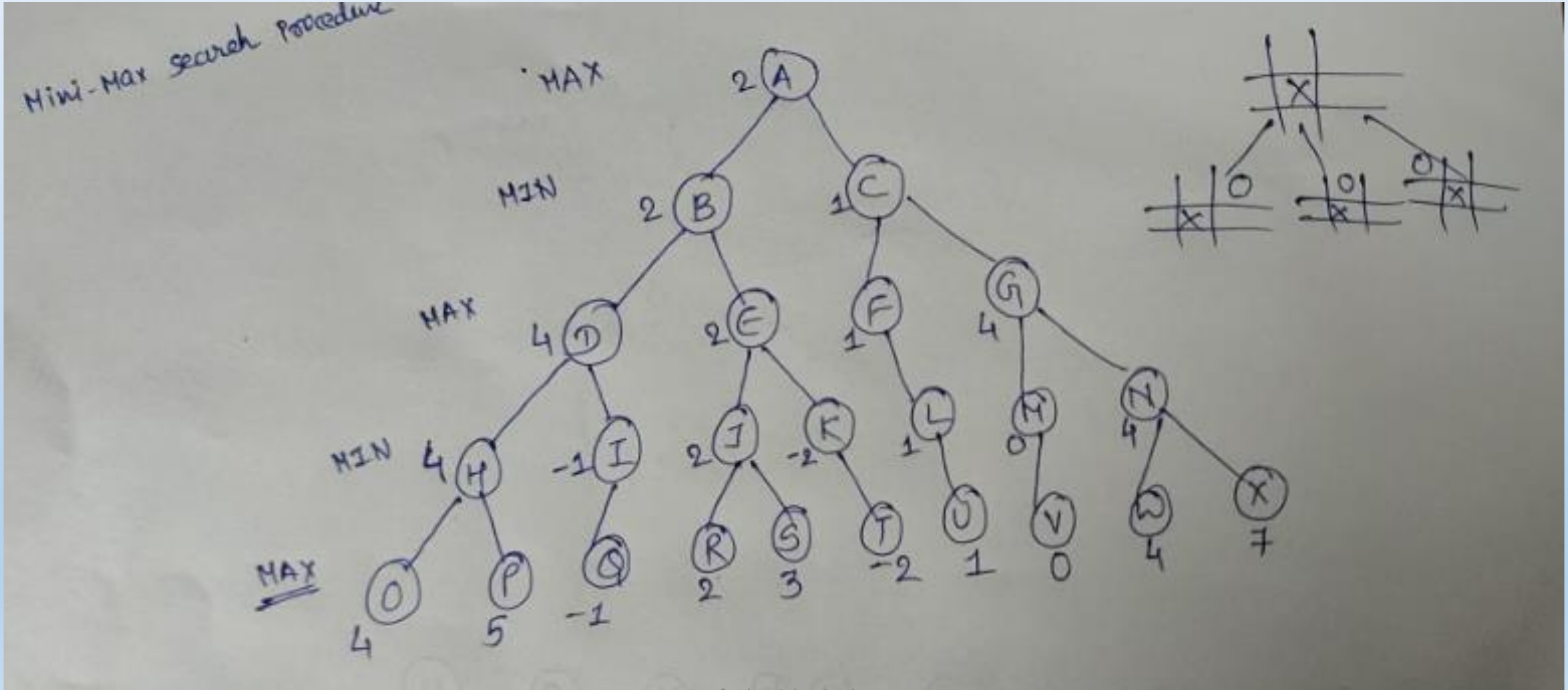
E(n): total Evaluation for state n



Back Values: Tic-Tac-Toe Game

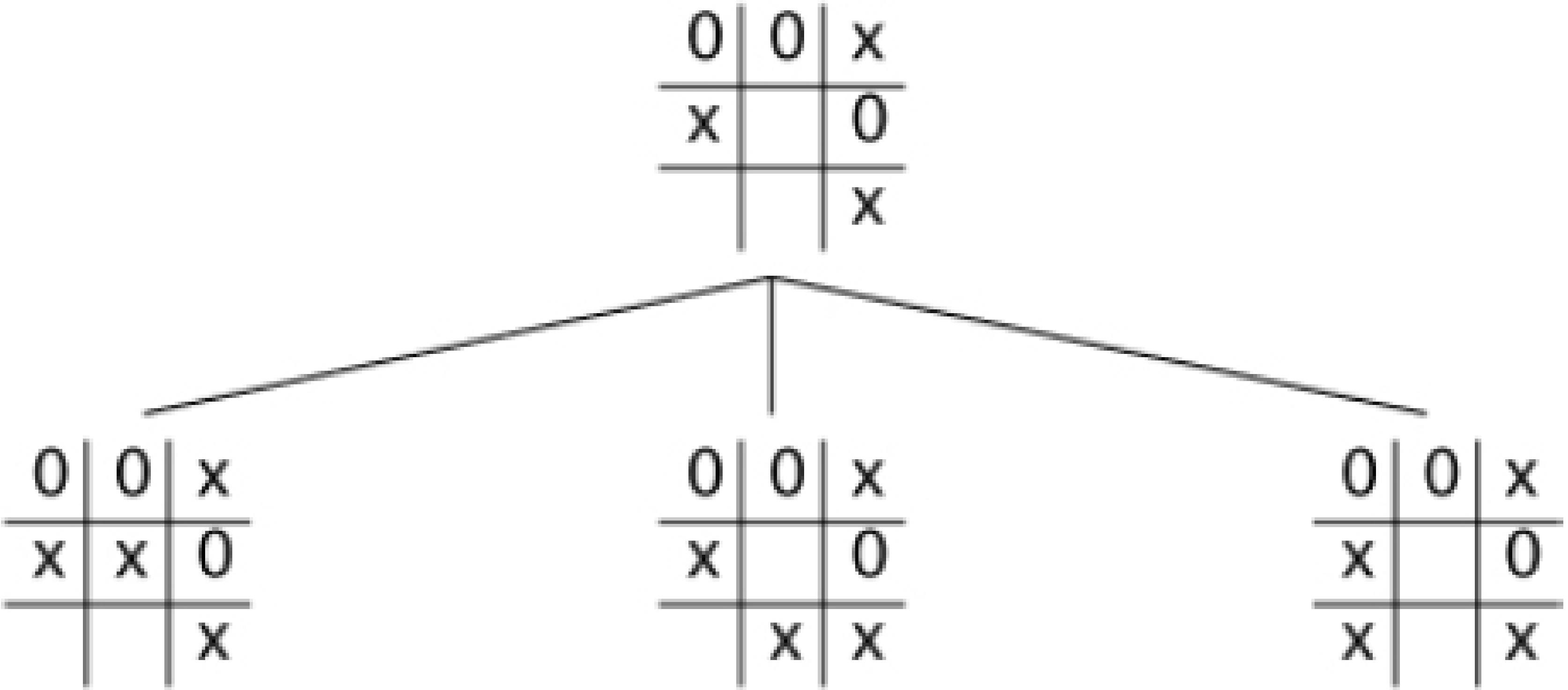


Mini-Max Search Procedure: Tic-Tac-Toe Game

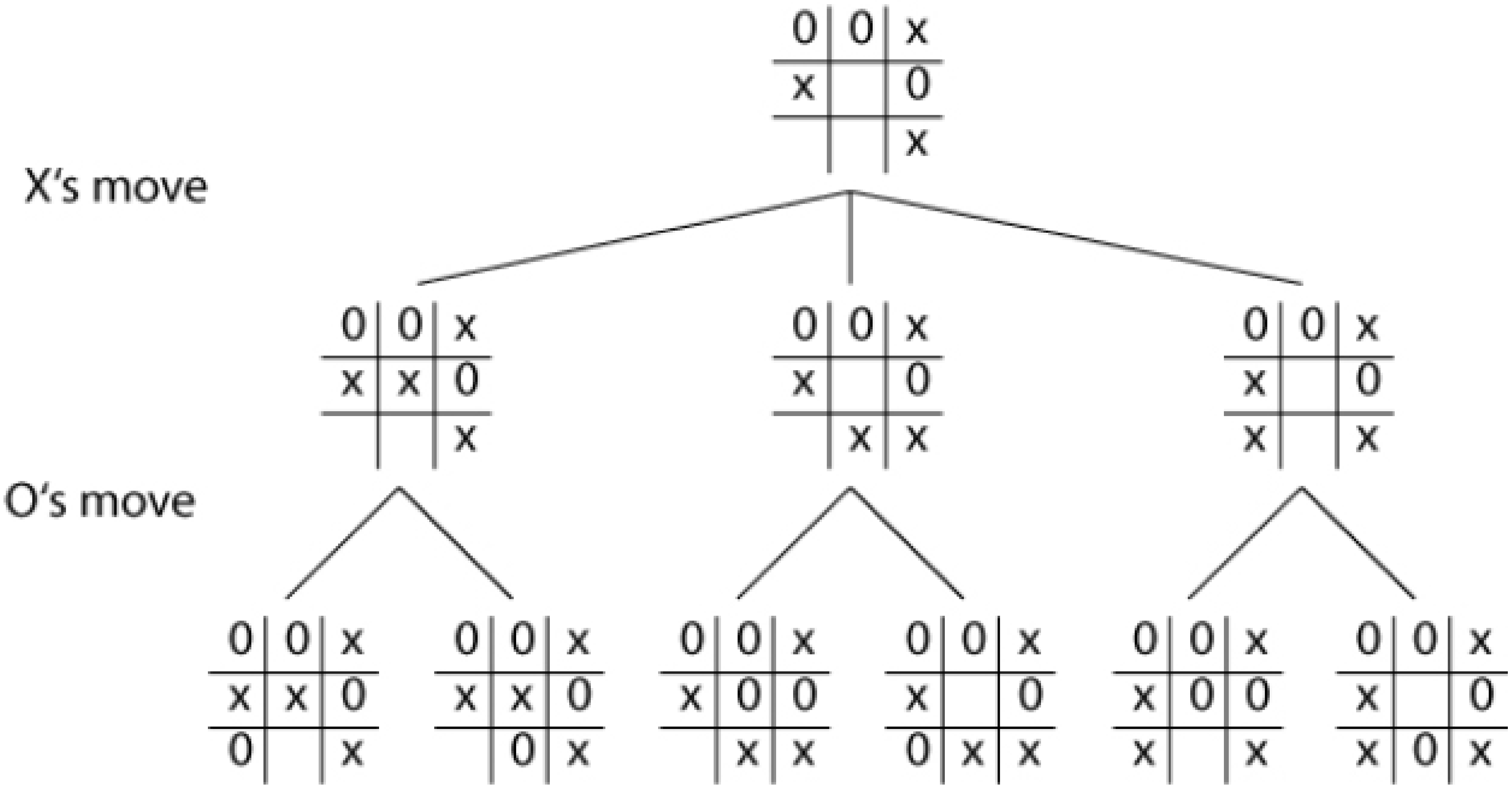


- **Start: X's Moves**

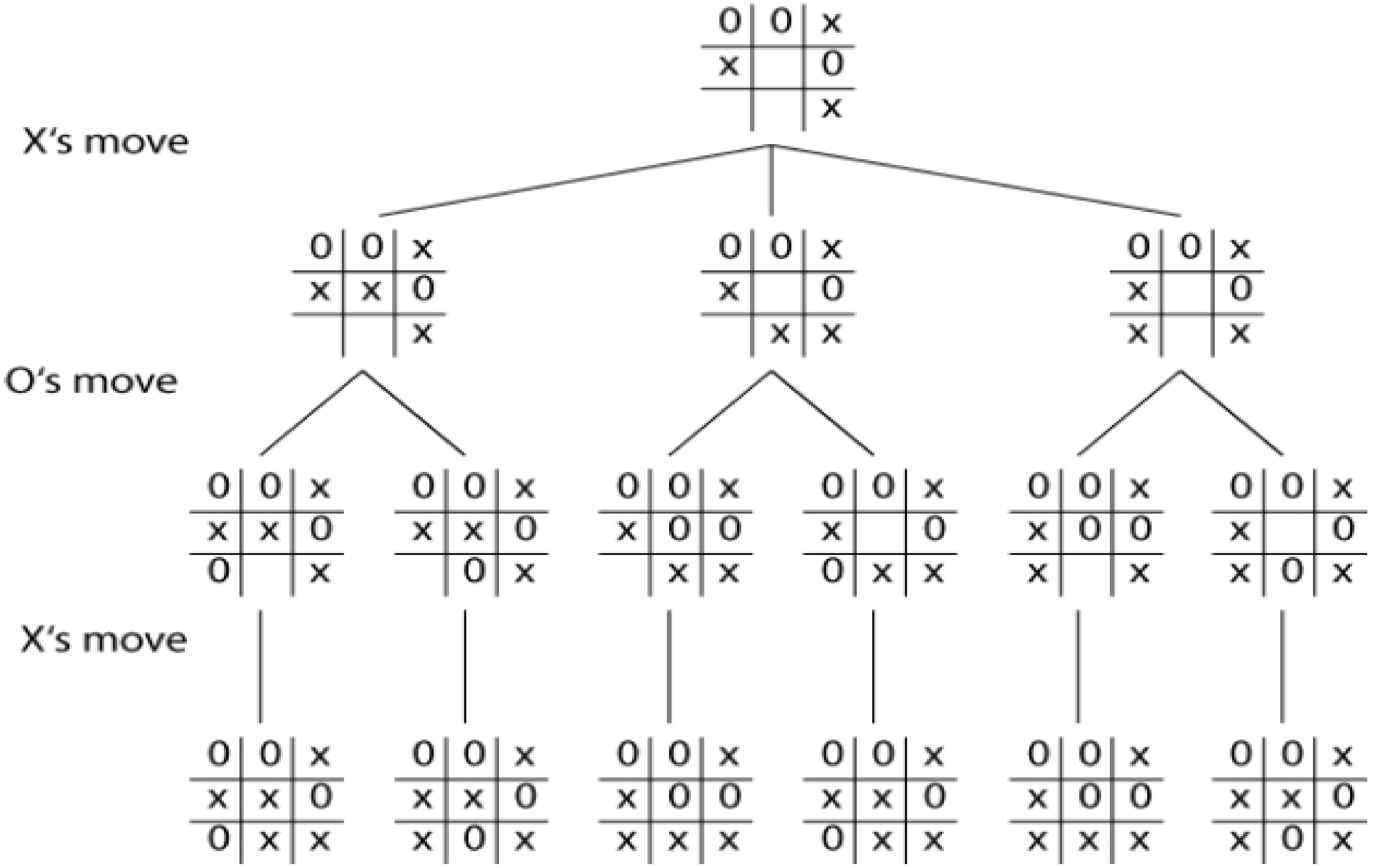
X's move



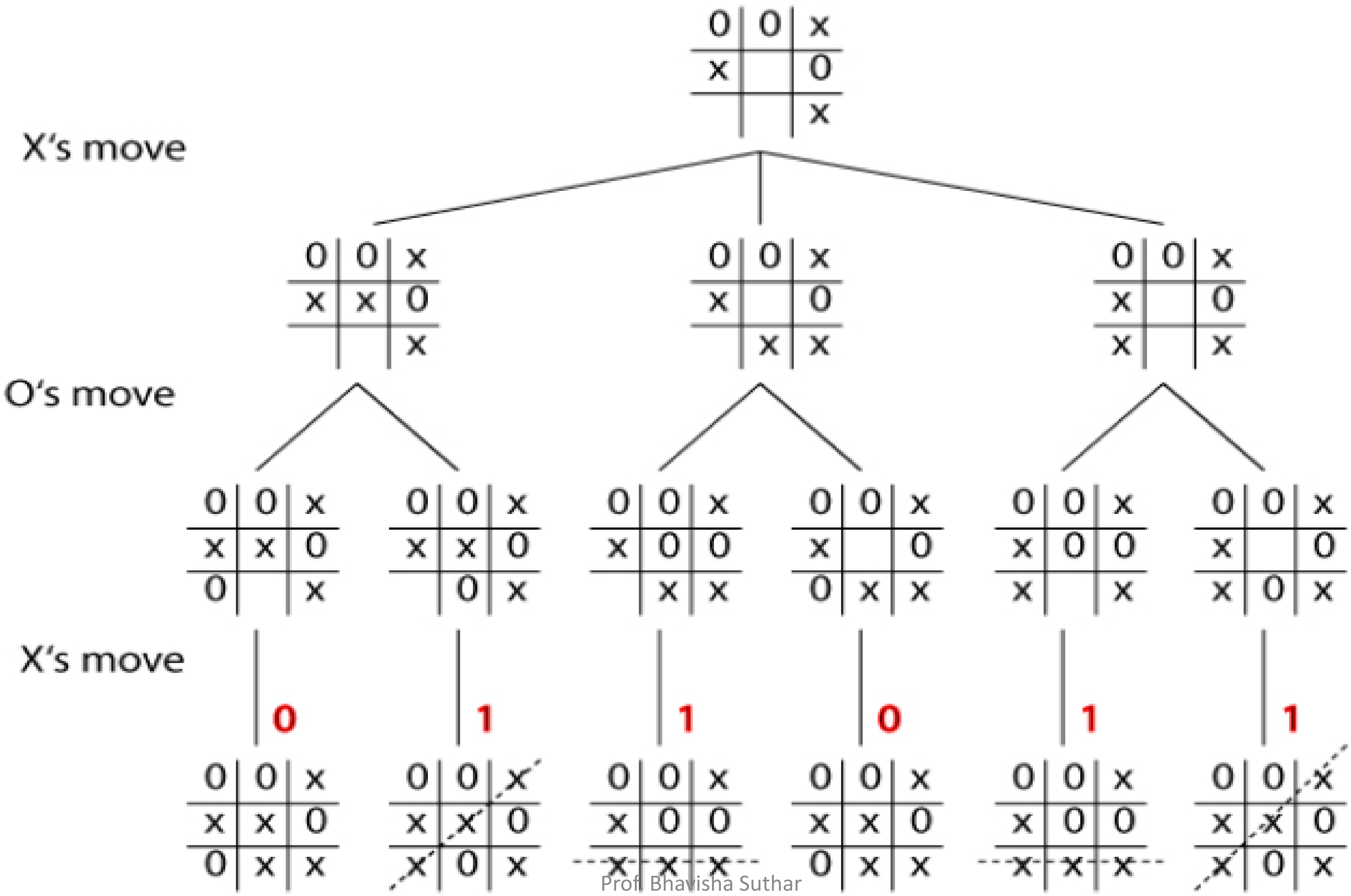
- **Next: O's Moves**



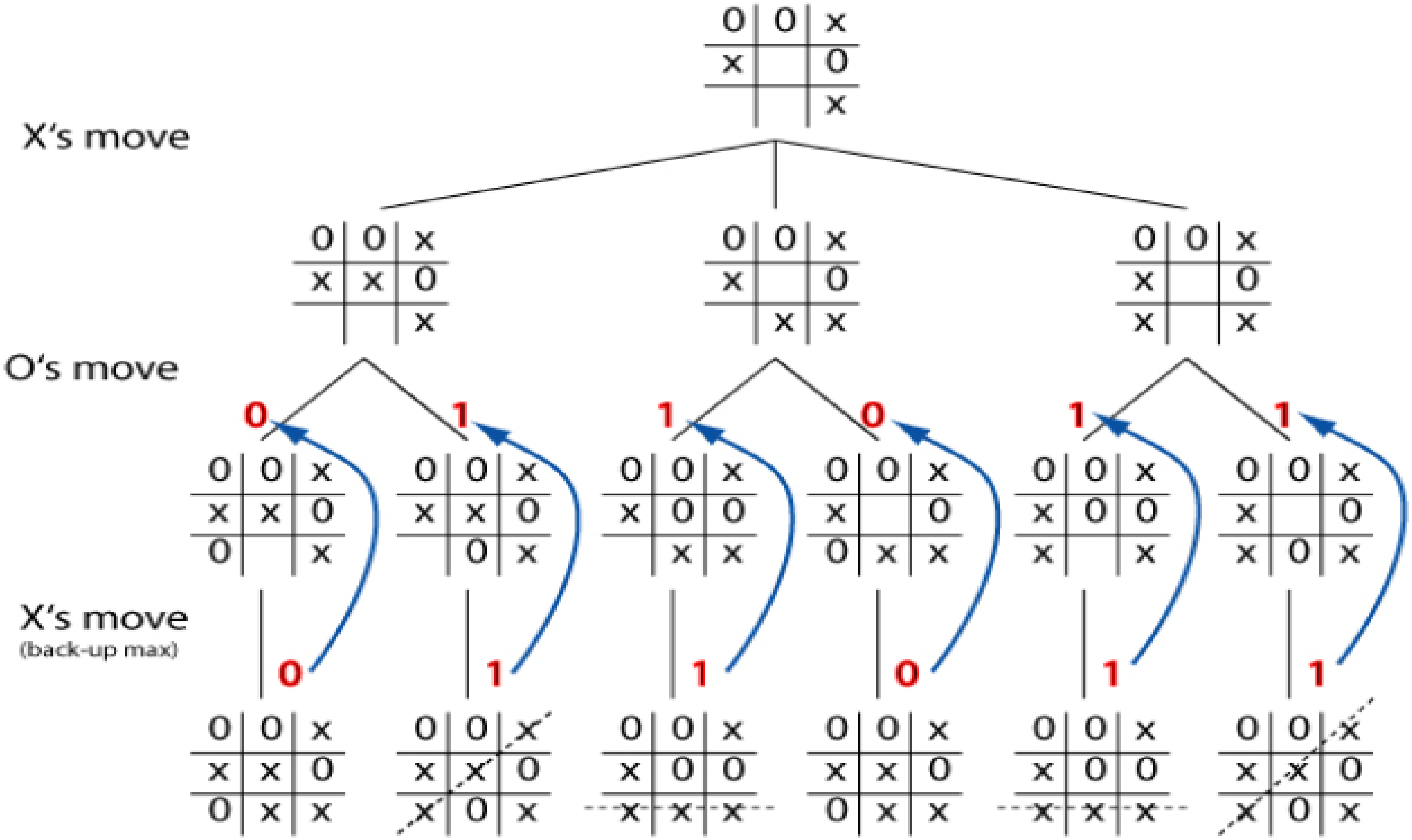
Again: X's moves

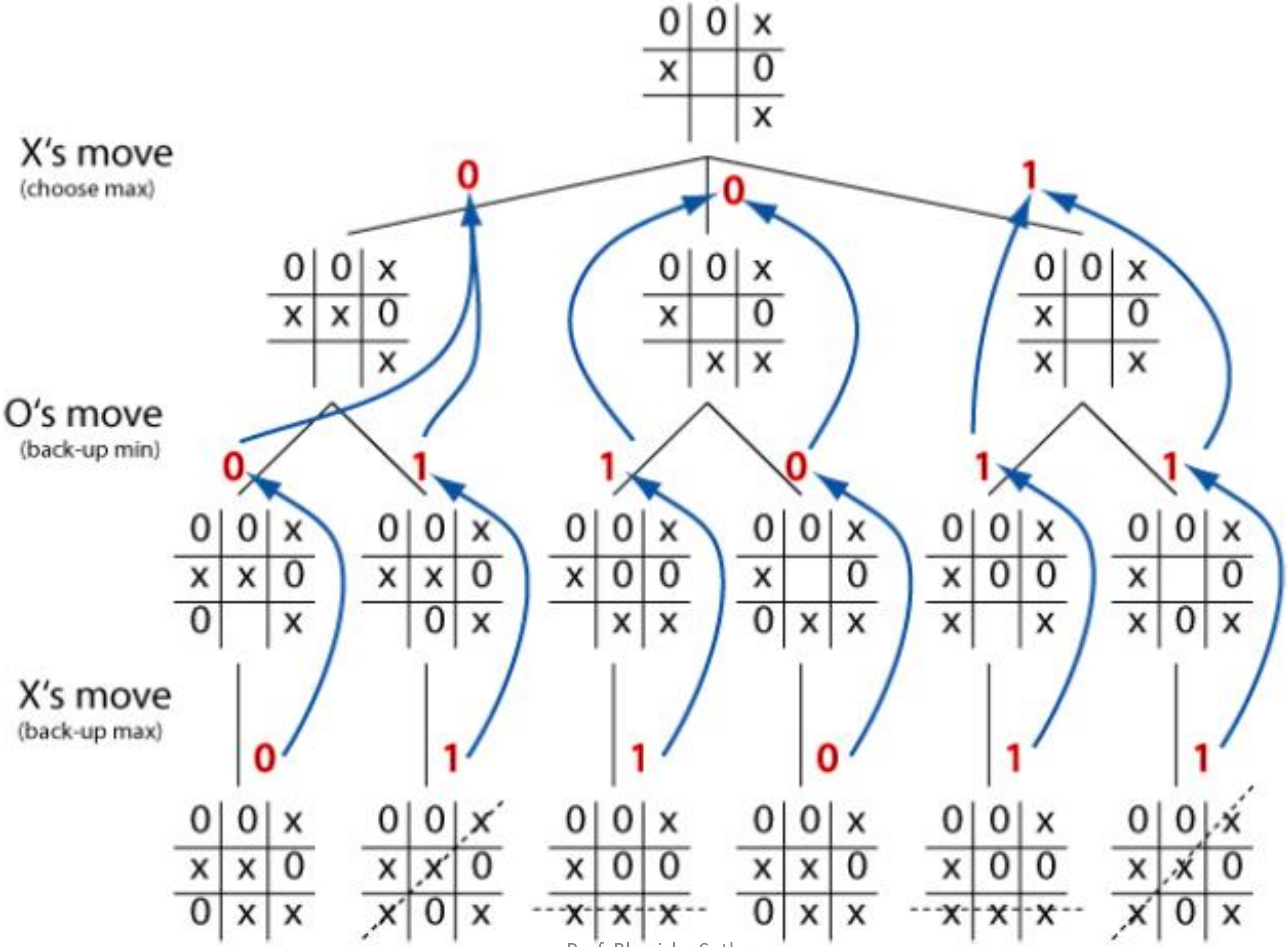


- Criteria '+1' for a Win, '0' for a Draw

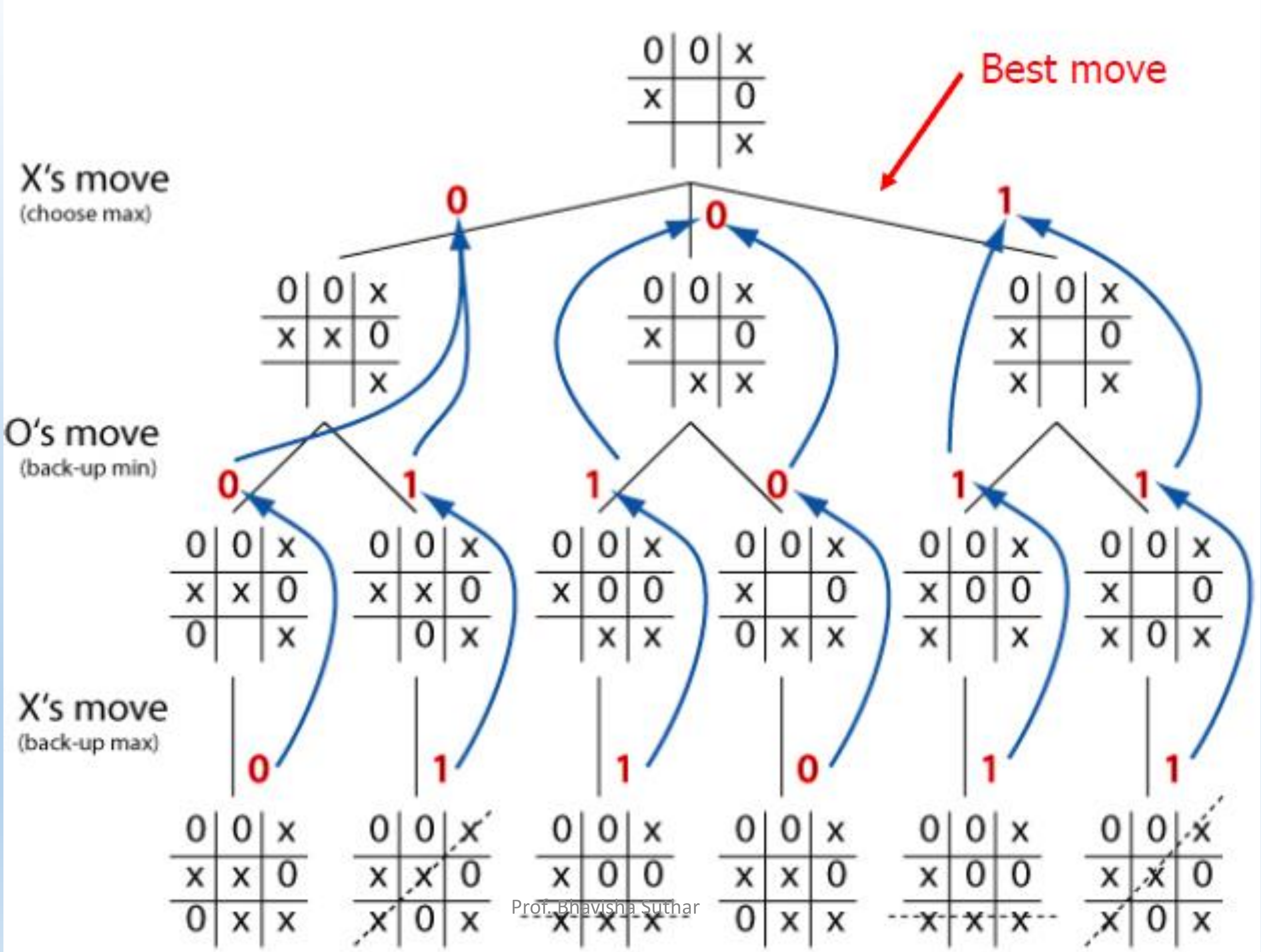


Up : One Level





X's move
(choose max)
O's move
(back-up min)
X's move
(back-up max)


Best move
X's move
(choose max)
O's move
(back-up min)
X's move
(back-up max)

Conclusion: Mini Max Search Procedure

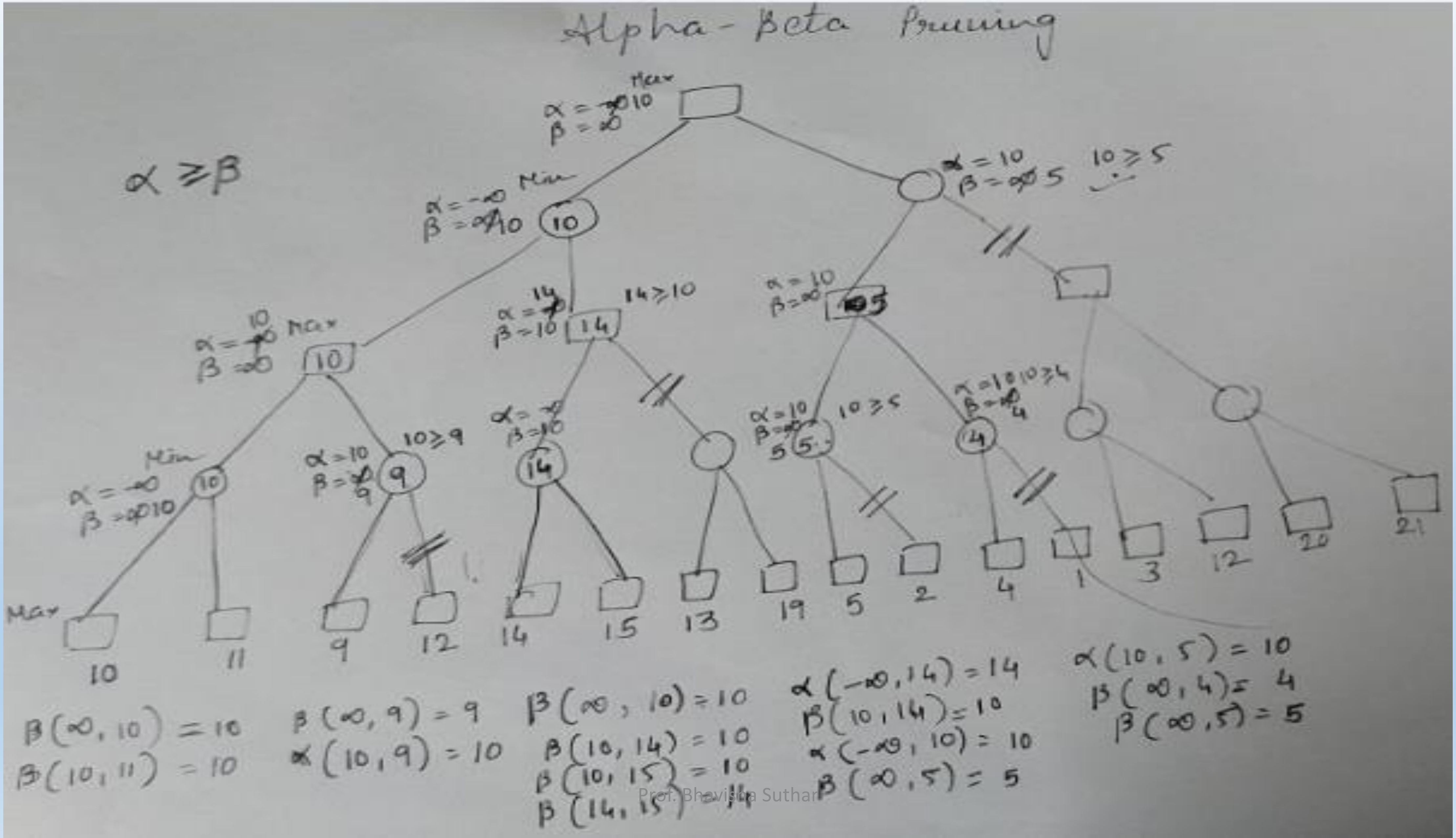
- Time complexity? O(bm), b is the branching degree, m is depth of tree.
- Space complexity? O(bm) (depth-first exploration)

Tic-Tac-Toe

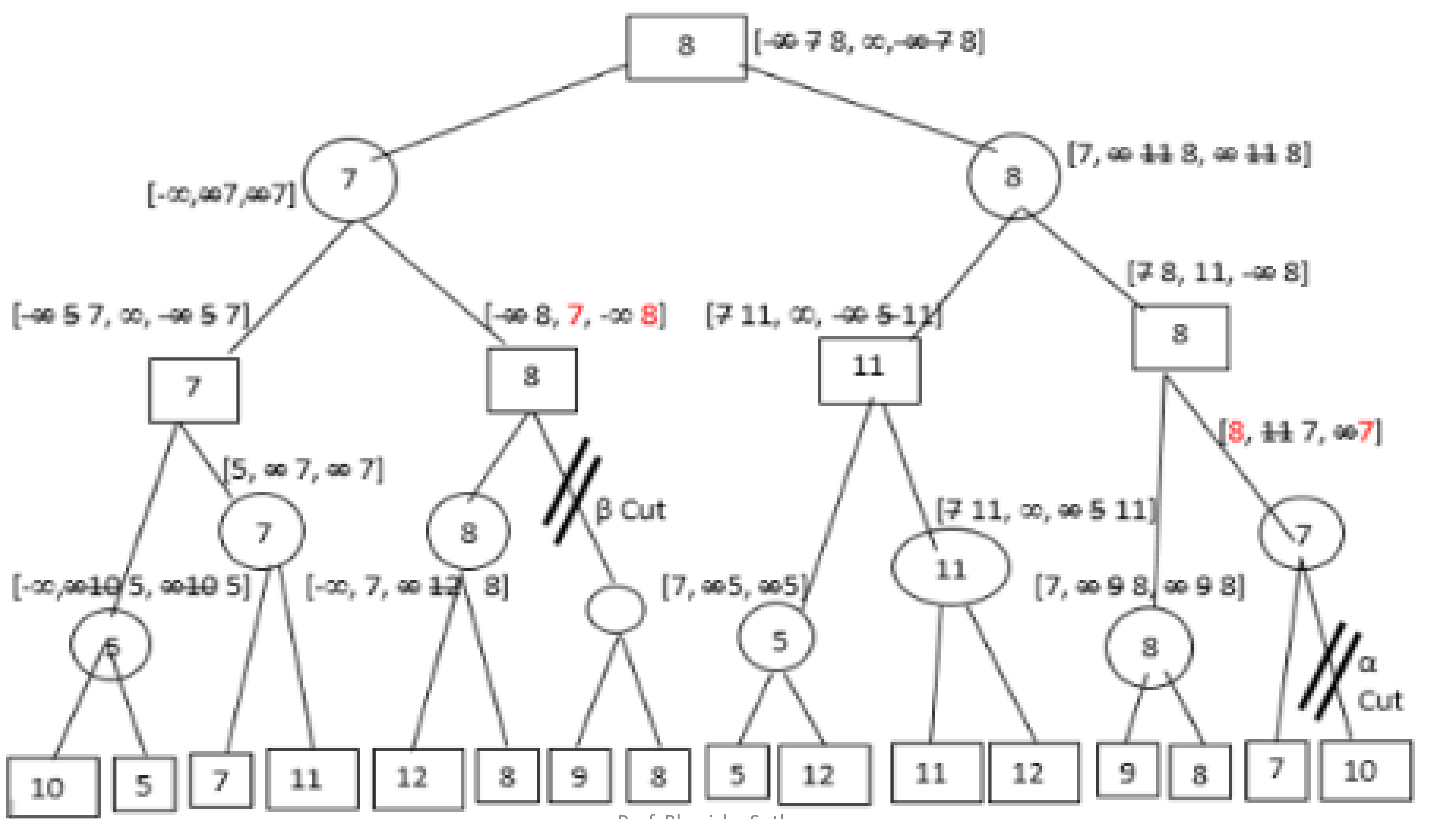
- 9! = 362,880 (Computer goes first)
- 8! = 40,320 (Computer goes second)

For chess, for "reasonable" games exact solution infeasible

Alpha-Beta Pruning: Tic-Tac-Toe Game



Alpha-Beta Pruning: Tic-Tac-Toe Game



Reference for code:

https://www.google.com/search?q=solved+tic+tac+toe+using+mini+max+search+procesure&rlz=1C1GIWA_enIN771IN771&oq=solved+tic+tac+toe+using+mini+max+search+procesure&aqs=chrome..69i57j33i10i160.19639j1j7&sourceid=chrome&ie=UTF-8#kpvalbx=_e-eYX6XDB5qZ4-EPtaWoaA24