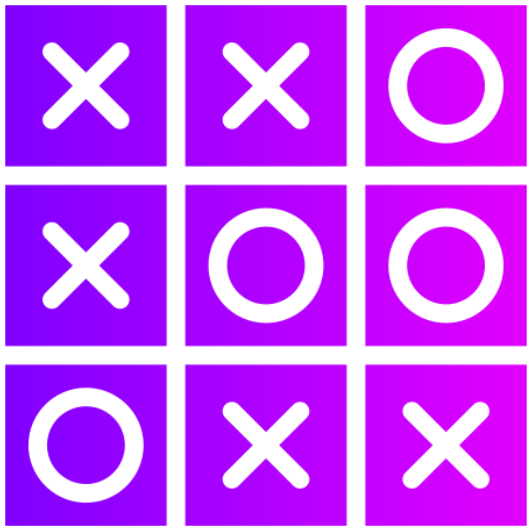
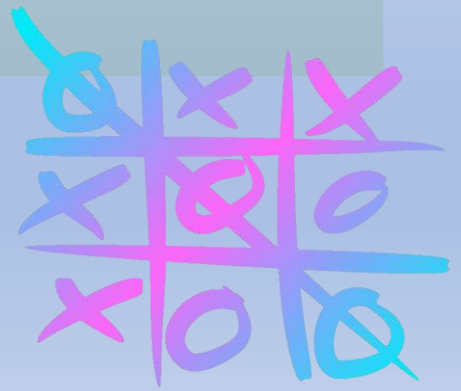


Tic-Tac-Toe with Artificial Intelligence

S M Asiful Islam Saky

Agenda

1. Playing Game
2. Tic-Tac-Toe
3. A vs. B
4. How to play?
5. Minimax Algorithm
6. Game tree for Tic-Tac-Toe
7. Python Implementation of Tic-Tac-Toe
8. Conclusion
9. Explore Tic-Tac-Toe



Playing Game

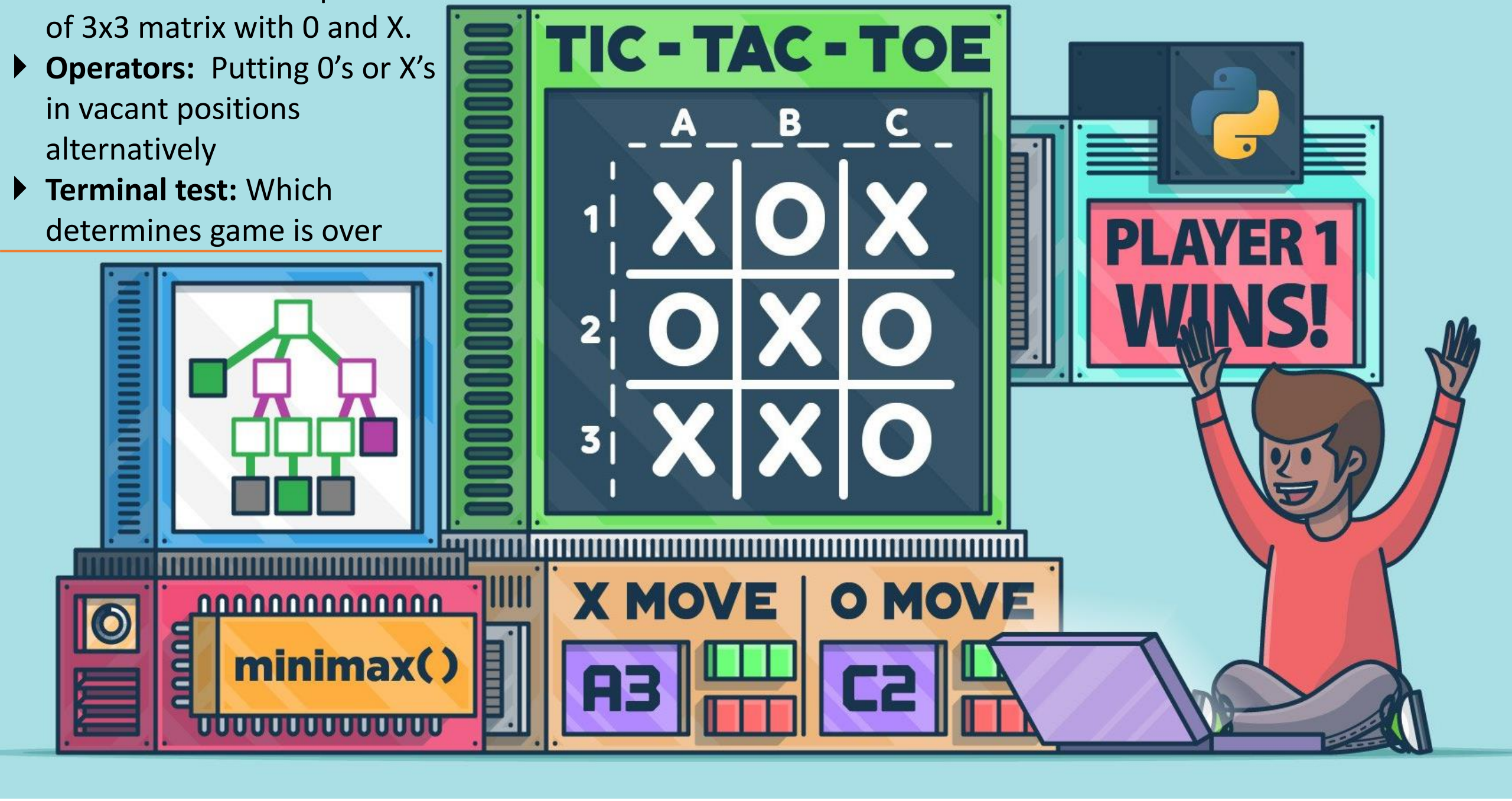
Games vs Search Problems

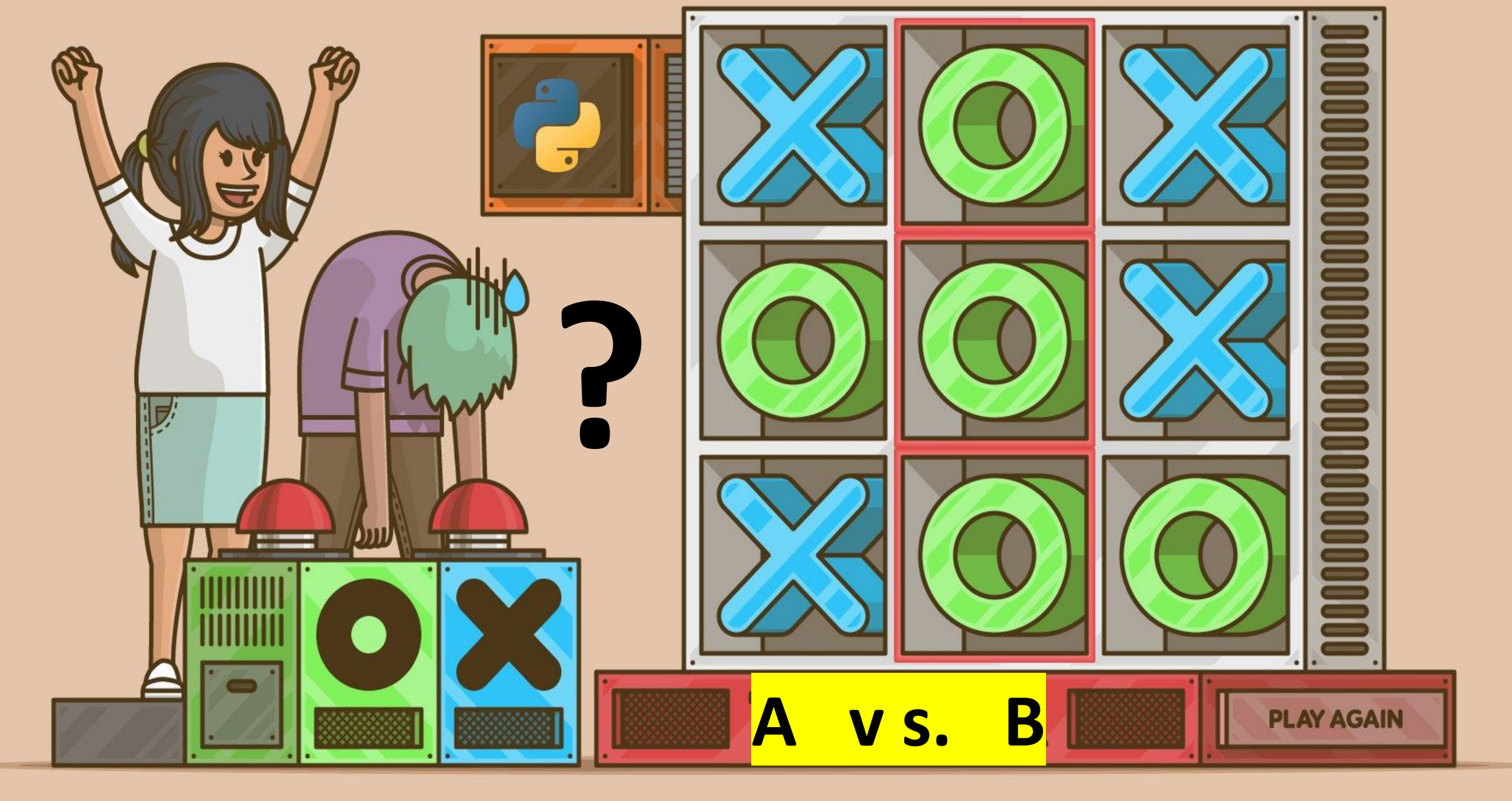
- "Unpredictable" opponent : specifying a move for every possible opponent reply.
- Time limits : unlikely to find goal, must approximate.

Game Playing Strategy

- Maximize winning possibility assuming that opponent will try to minimize (Minimax Algorithm)
- Ignore the unwanted portion of the search tree (Alpha Beta Pruning)
- Evaluation(Utility) Function(A measure of winning possibility of the player)

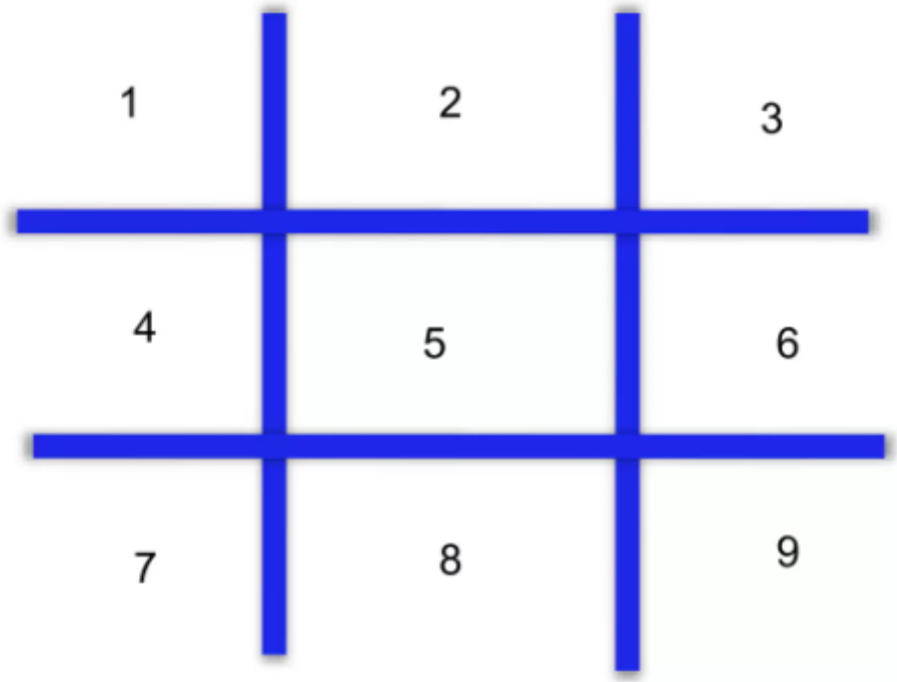
- **Initial State**: Board position of 3x3 matrix with 0 and X.
- **Operators:** Putting 0's or X's in vacant positions alternatively
- **Terminal test:** Which determines game is over




?
A vs. B
PLAY AGAIN

Initial State

Initially all 1 to 9 places of 3x3 grid are empty



Goal State

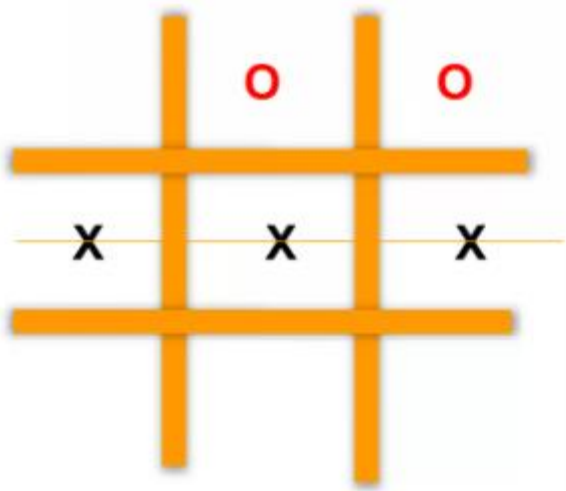
PLAYER X WIN

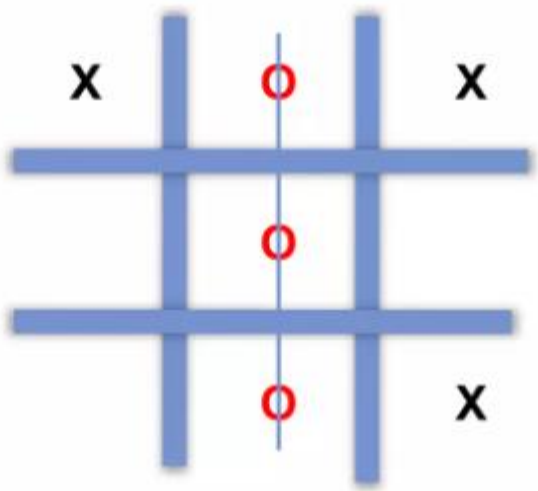
3 X's in a column , row or diagonal

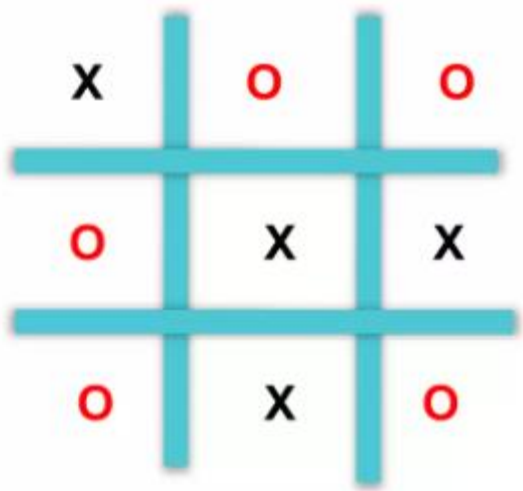
PLAYER O WIN

3 O's in a column , row or diagonal

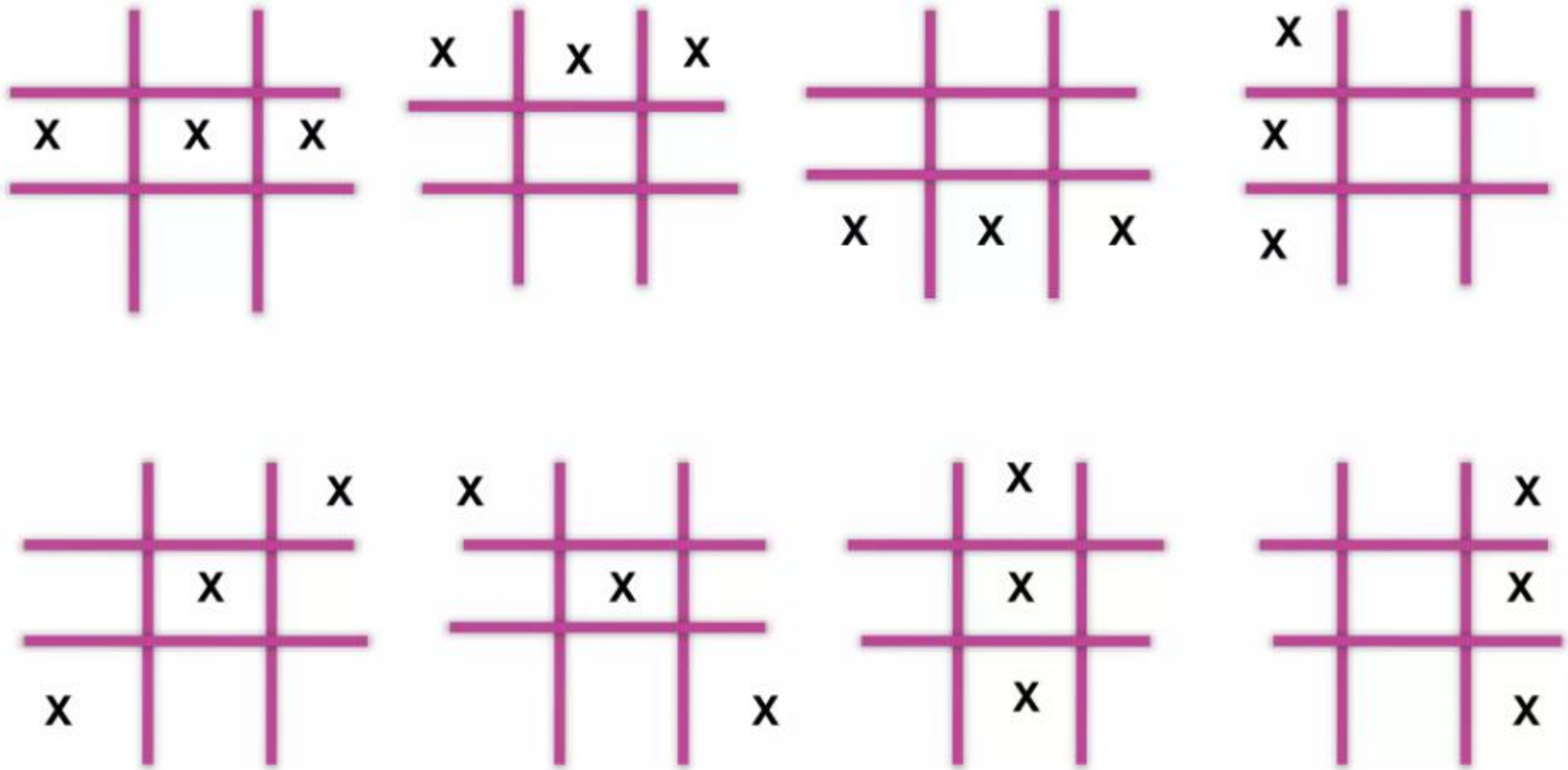
DRAW



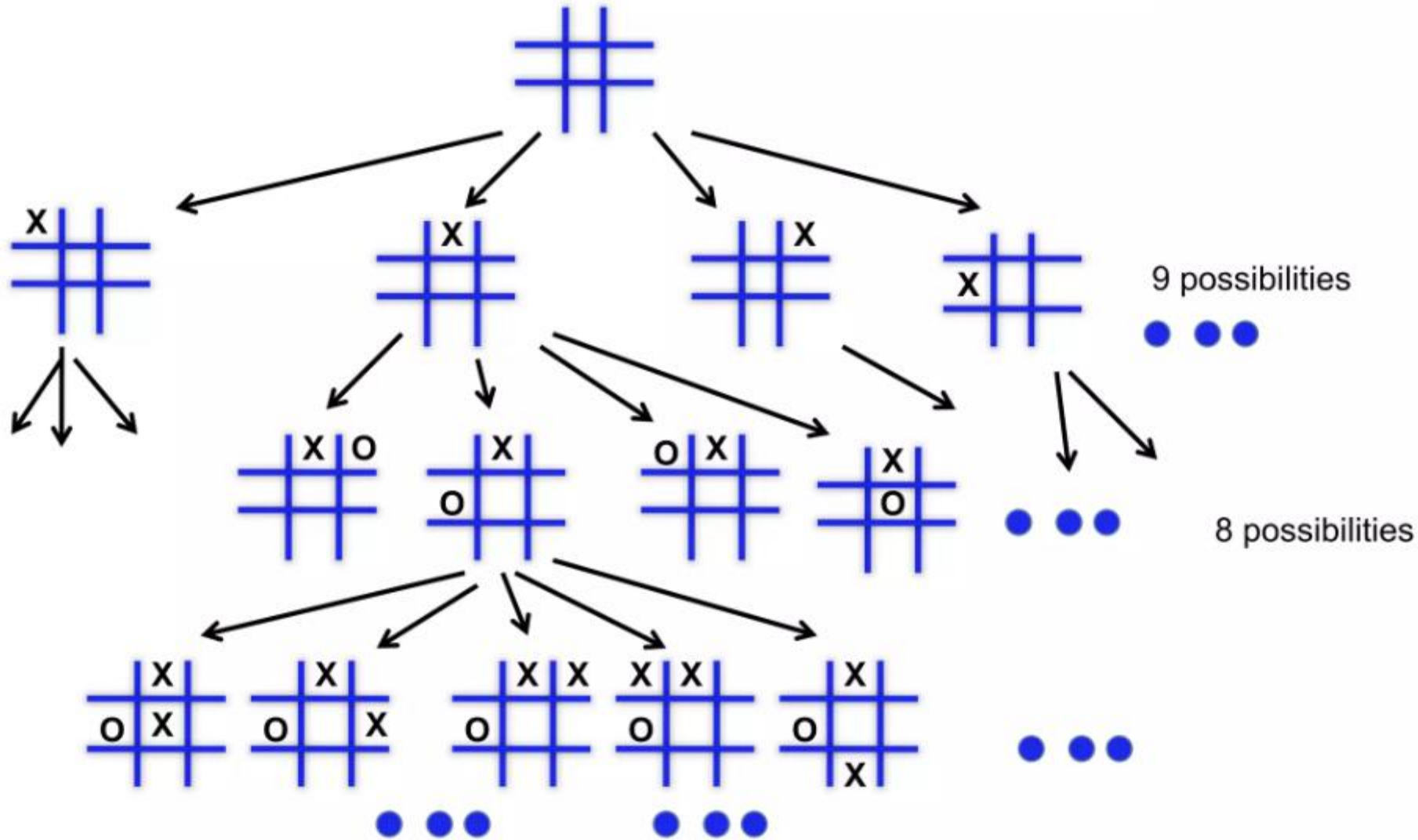


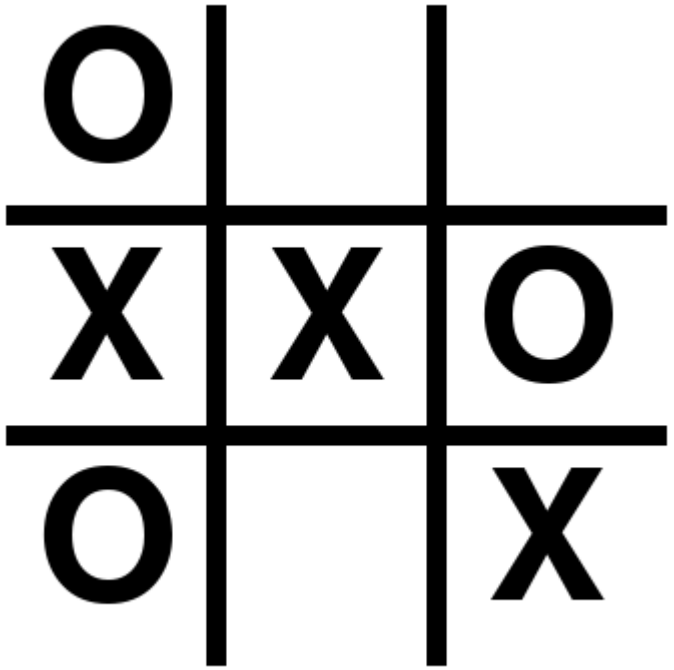


Winning Conditions



State Space Tree

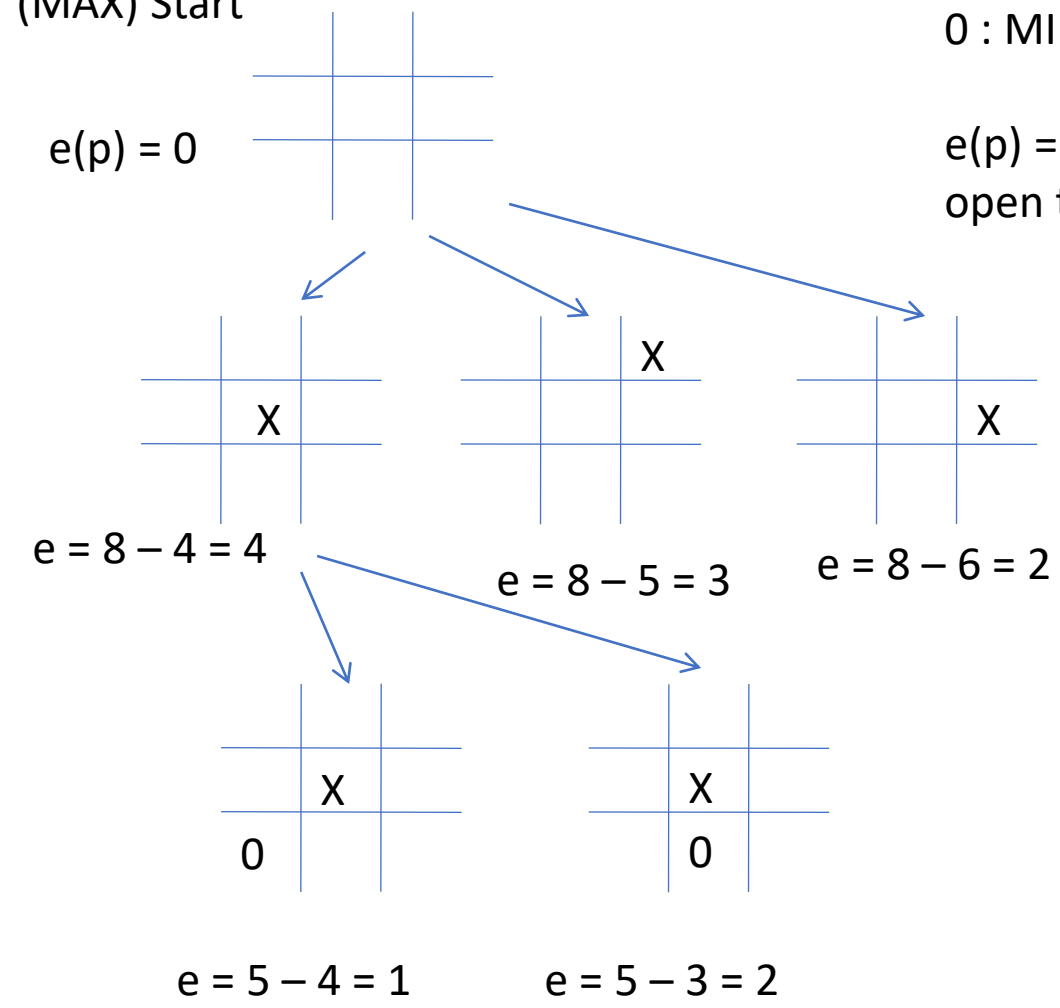




- e(p) = 6 - 5 = 1
- **Utility function**:

  e(p) = (No. of complete rows, columns or diagonals are still open for player) – (No. of complete rows, columns or diagonals are still open for opponent)

Tic-Tac-Toe



X : MAX player
0 : MIN player

e(p) = (rows + cols + diagonals open to ‘X’) – (Same to ‘0’)

X’s Turn

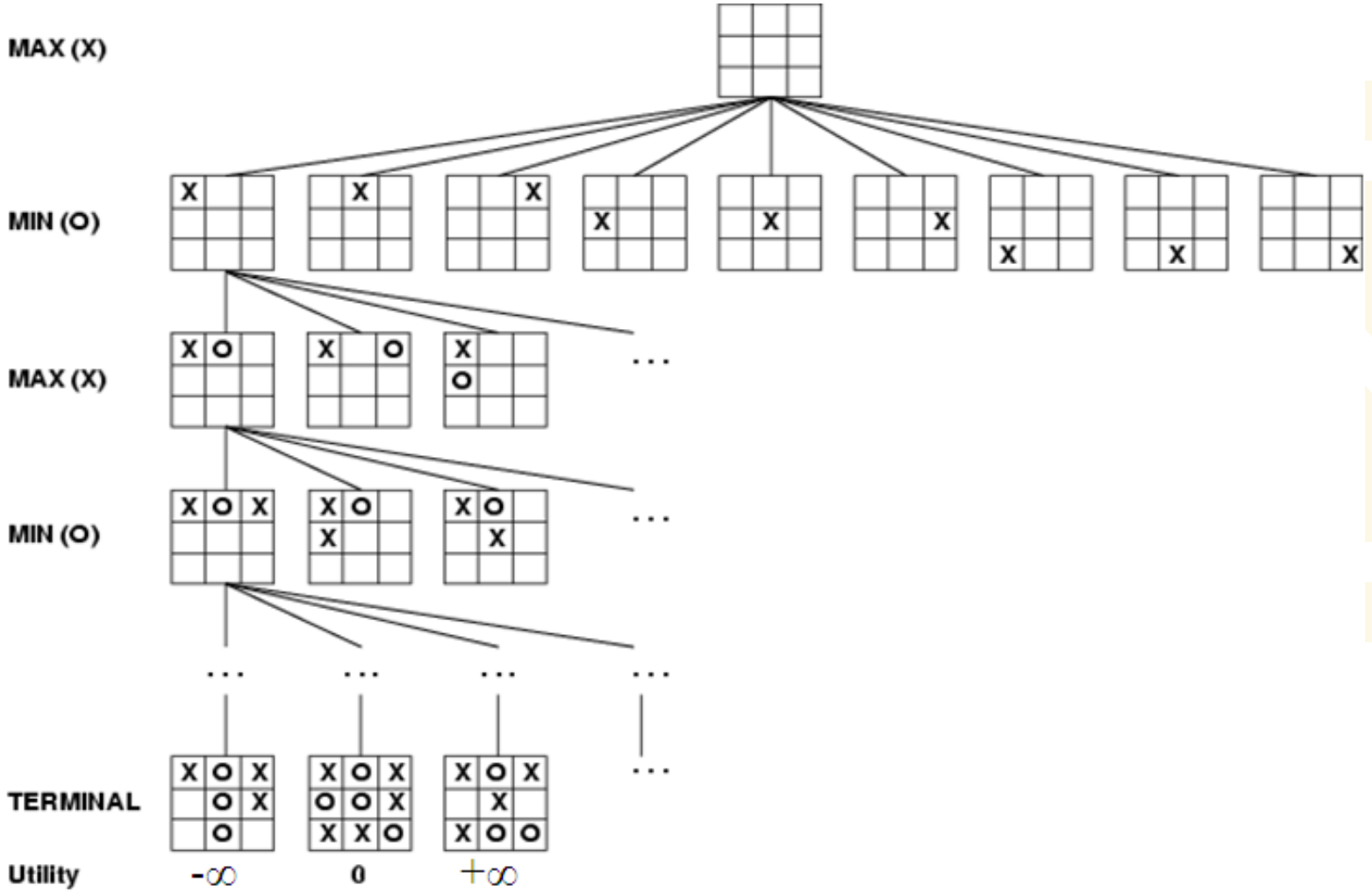
0’s Turn

Courtesy : CS621-Artificial Intelligence , 2007, Prof. Pushpak Bhatacharya

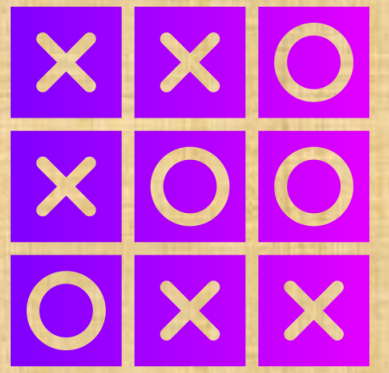
Minimax Algorithm

- Generate the game tree
- Apply the utility function to each terminal state to get its value
- Use these values to determine the utility of the nodes one level higher up in the search tree
  - From bottom to top
  - For a max level, select the maximum value of its successors
  - For a min level, select the minimum value of its successors
- From root node select the move which leads to highest value

Game tree for Tic-Tac-Toe



Courtesy : Artificial Intelligence and Soft Computing. Behavioural and Cognitive Modelling of the Human Brain



Python

Implementation of

Tic-Tac-Toe

```
New Game!
---------------
|   ||   ||   |
---------------
|   ||   ||   |
---------------
|   ||   ||   |
---------------
Choose which player goes first - X (You - the petty human) or O(The mighty AI): X
Oye Human, your turn! Choose where to place (1 to 9): 1
---------------
| X ||   ||   |
---------------
|   ||   ||   |
---------------
|   ||   ||   |
---------------
Possible moves = [2, 3, 4, 5, 6, 7, 8, 9]
Move values = [-0.175858, 0.01031, -0.051866, 0.250661, -0.008078, 0.013969, -0.002939, -0.00299]
AI plays move: 5
---------------
| X ||   ||   |
---------------
|   || O ||   |
---------------
|   ||   ||   |
---------------
Oye Human, your turn! Choose where to place (1 to 9): 5
Block is not empty, ya blockhead! Choose again: 4
---------------
| X ||   ||   |
---------------
| X || O ||   |
---------------
|   ||   ||   |
---------------
Possible moves = [2, 3, 6, 7, 8, 9]
Move values = [0.119088, -0.075291, -0.082223, 0.095653, -0.070901, -0.218476]
AI plays move: 2
---------------
| X || O ||   |
---------------
| X || O ||   |
---------------
|   ||   ||   |
---------------
```

Conclusion

The game of Tic Tac Toe with AI has become an immensely popular game. It has provided an exciting and entertaining game experience for many generations. With the right AI choices, even more sophisticated strategic approaches can be implemented and the game can become even more challenging and exciting. AI-driven Tic Tac Toe is a great way to engage your brain, invigorate your creativity, challenge your skills, and hone your problem-solving abilities. AI technology is constantly advancing, and future AI-driven Tic Tac Toe games will present even more intriguing puzzles and strategies.

TIC TAC TOE A.I.

```
(x):
n x+5

write(ast):
ename = getNodename()
el=symbol.sym_name.get(int(ast[0]),ast[0])
nt '     %s [label="%s' % (nodename, label),
 isinstance(ast[1], str):
 if ast[1].strip():
    print '= %s"];' % ast[1]
```


ANY
QUESTIONS?

References

- D. E. Knuth and R. W. Moore. An analysis of alpha-beta pruning. Artificial Intelligence, 6:293–326, 1975
- Rich, E. and Knight, K., *Artificial Intelligence, McGraw-Hill, New* York, 1991.
- Nilson, J. N., *Principles of Artificial Intelligence, Morgan-Kaufmann,*

  San Mateo, CA, pp. 112-126,1980.
- Russel, S. and Norvig, P., *Artificial Intelligence: A Modern Approach,*

  Prentice-Hall, Englewood Cliffs, NJ, 1995.
- Amit Konar , *Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain,* CRC Press 2000.


Thank You

S M Asiful Islam Saky