Artificial Intelligence (AI-2002)

Chapter 5: Adverserial Search

Content

- Adversarial Search
- Game Optimal Decision
- Min-max
- αβ-Pruning

Adversarial Search

- There might be some situations where more than one agent is searching for the solution in the same search space, and this situation usually occurs in game playing.
- The environment with more than one agent is termed as **multi-agent environment**
- Search in which two or more players with conflicting goals are trying to explore the same search space for the solution is called as an **Adversarial search**, often known as **Games**.

Types of Games in AI

- Games are modeled as a Search problem and evaluation function, that helps to solve games.

	Deterministic	Non-deterministic
Perfect information	Chess,	Backgammon, monopoly
Imperfect information	Battleships	poker, scrabble

Perfect information: A game with the perfect information is that in which agents can look into the complete board.
Imperfect information: Agents do not have all information about the game
Deterministic games: Deterministic games are those games which follow a strict pattern and set of rules for the games.
Non-deterministic games: Non-deterministic are those games which have various unpredictable events

Zero-Sum Game

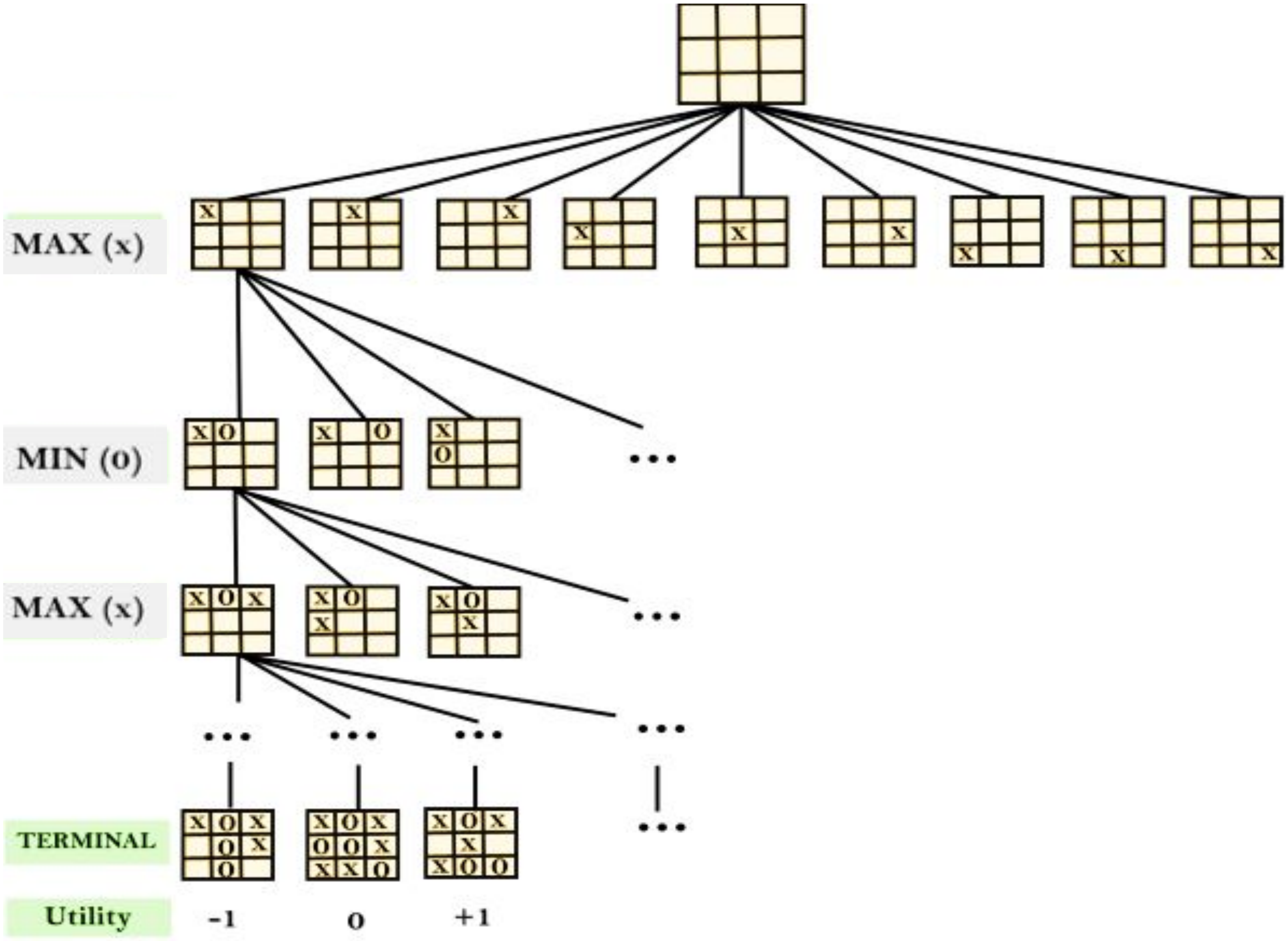
- Zero-sum games are adversarial search which involves pure competition.
- In Zero-sum game each agent's gain or loss of utility is exactly balanced by the losses or gains of utility of another agent.
- One player of the game try to maximize one single value, while other player tries to minimize it.

General games

- Agents have independent utilities
- Co-operative, compete or indifference

Game tree

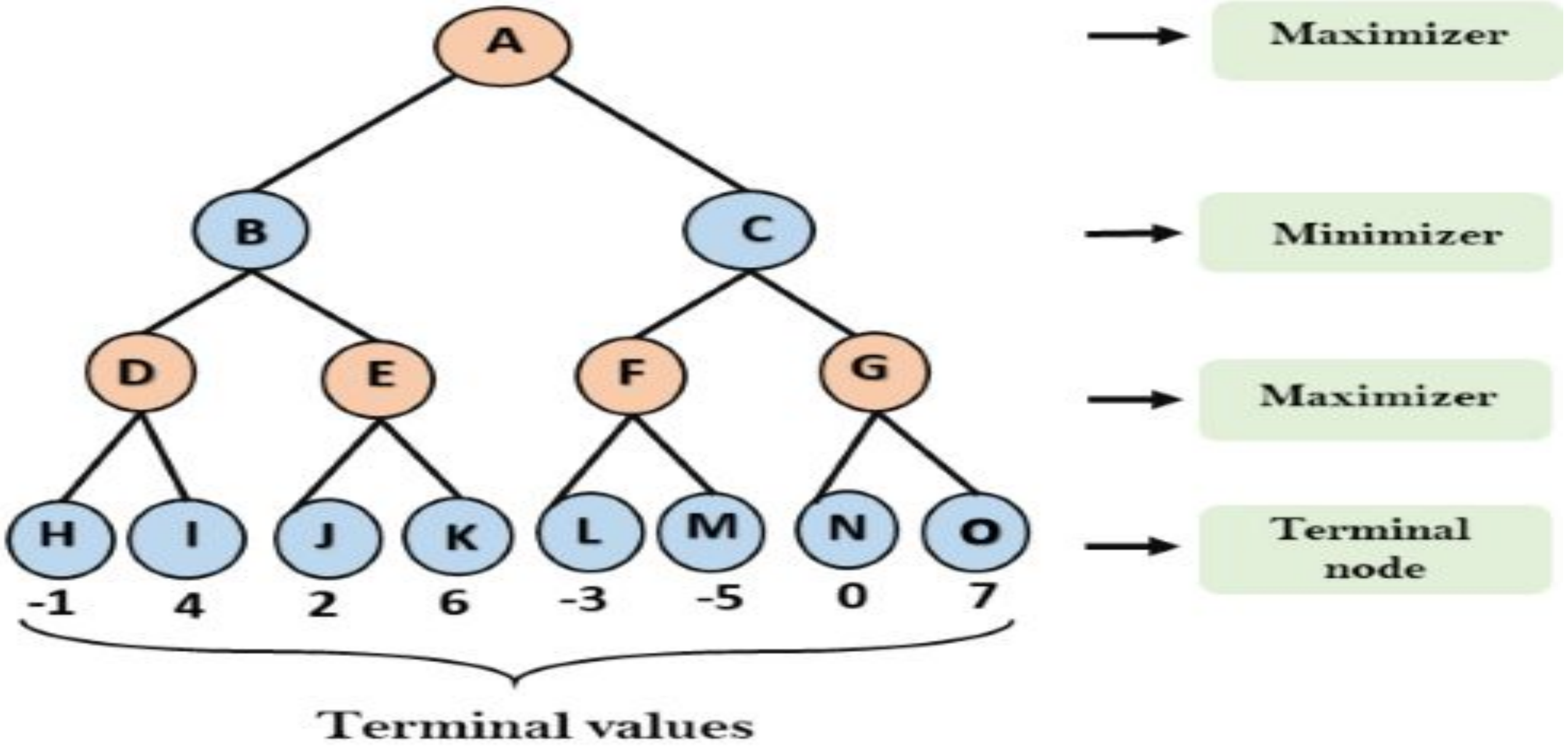
- A game tree is a tree where nodes of the tree are the game states and Edges of the tree are the moves by players. Game tree involves initial state, actions function, and result Function. (utility function)



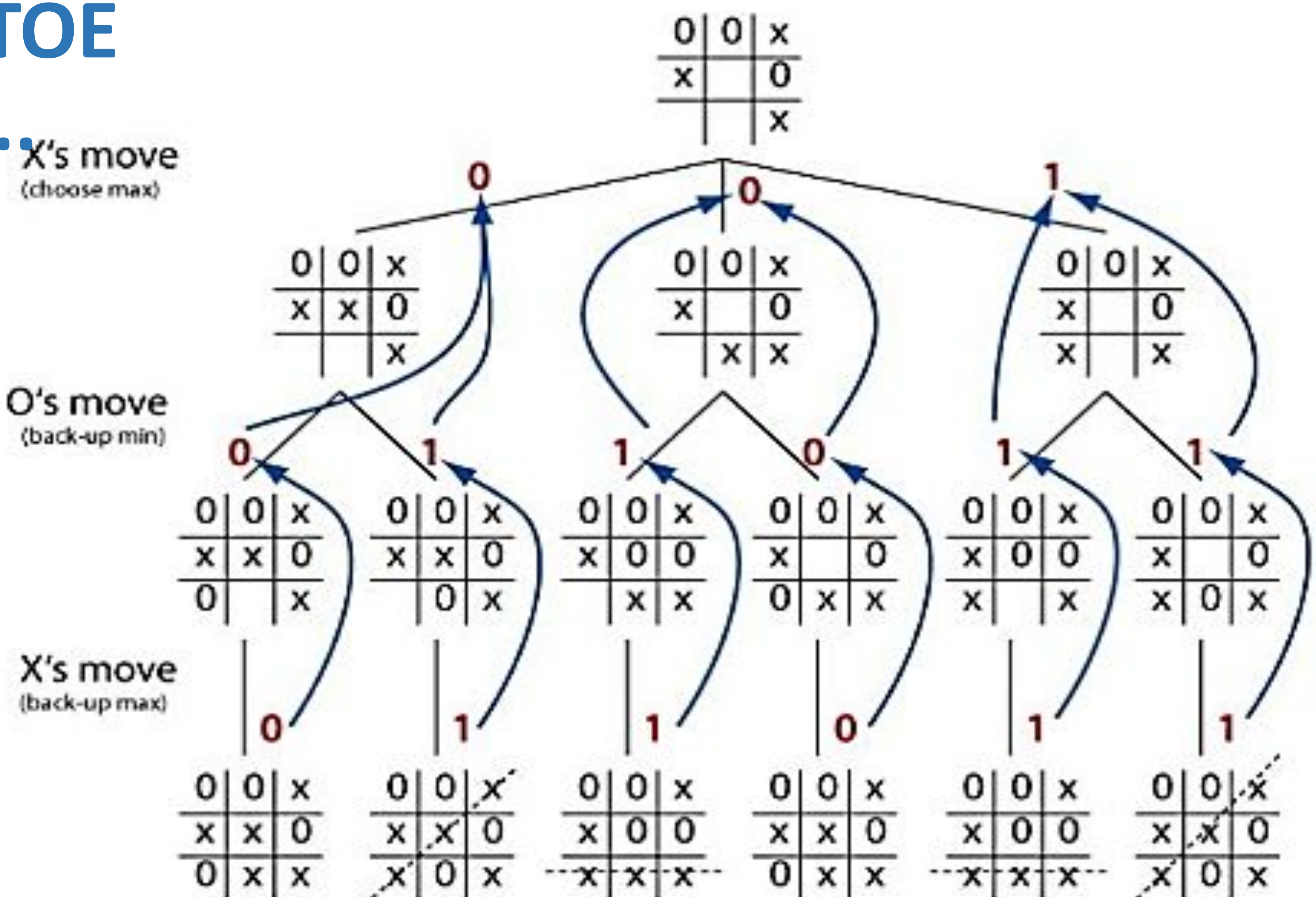
Min-Max Algorithm

- In this algorithm two players play the game, one is called MAX and other is called MIN.
- Both the players fight it as the opponent player gets the minimum benefit while they get the maximum benefit.
- Both Players of the game are opponent of each other, where MAX will select the maximized value and MIN will select the minimized value.
- The min-max algorithm performs a depth-first search algorithm for the exploration of the complete game tree.
- The min-max algorithm proceeds all the way down to the terminal node of the tree, then backtrack the tree as the recursion.

Min-Max Algorithm E.g………

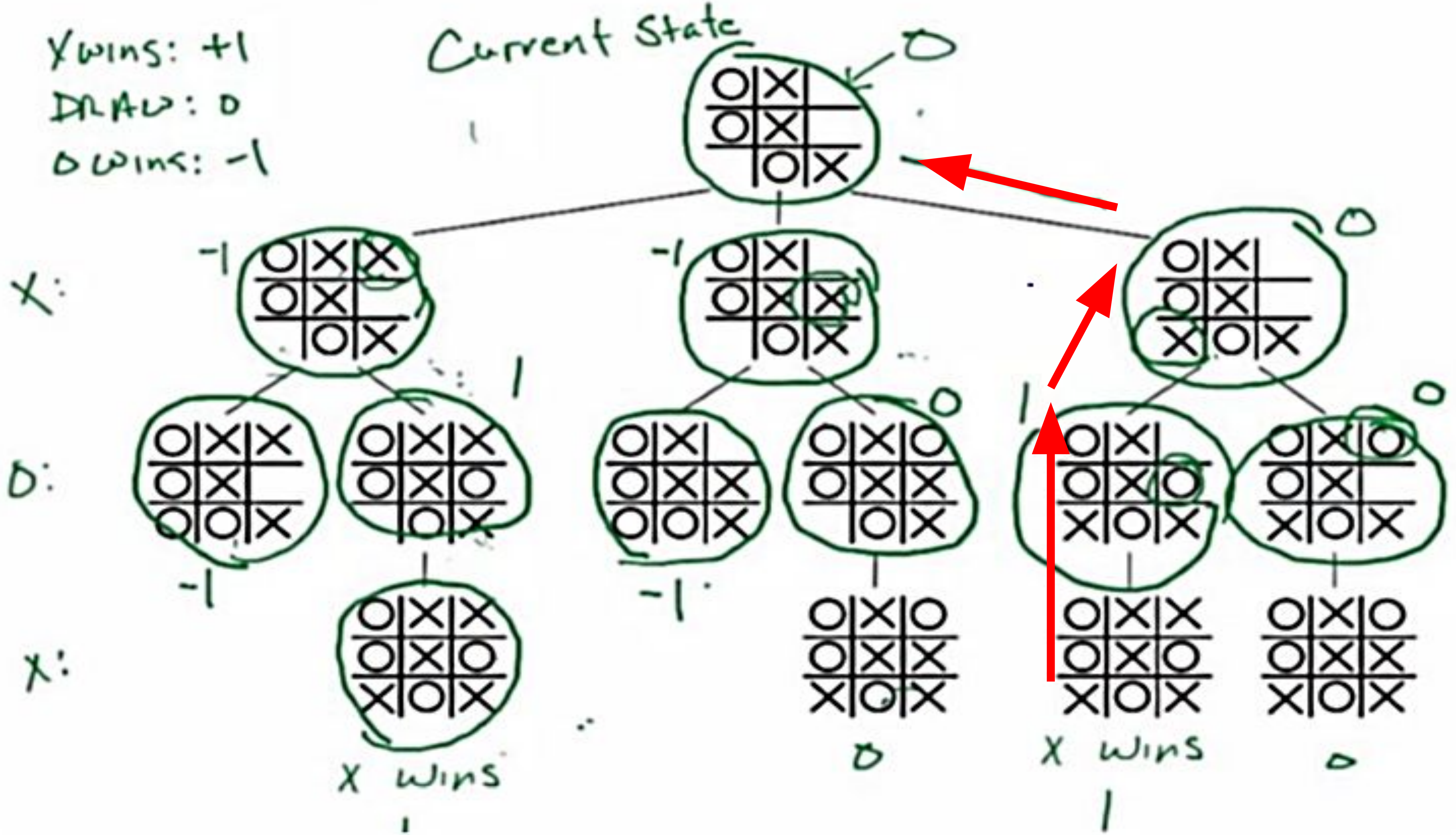


TIC-TAC TOE E.g………



Tic-Tac-Toe

X wins: +1
DRAW: 0
O wins: -1



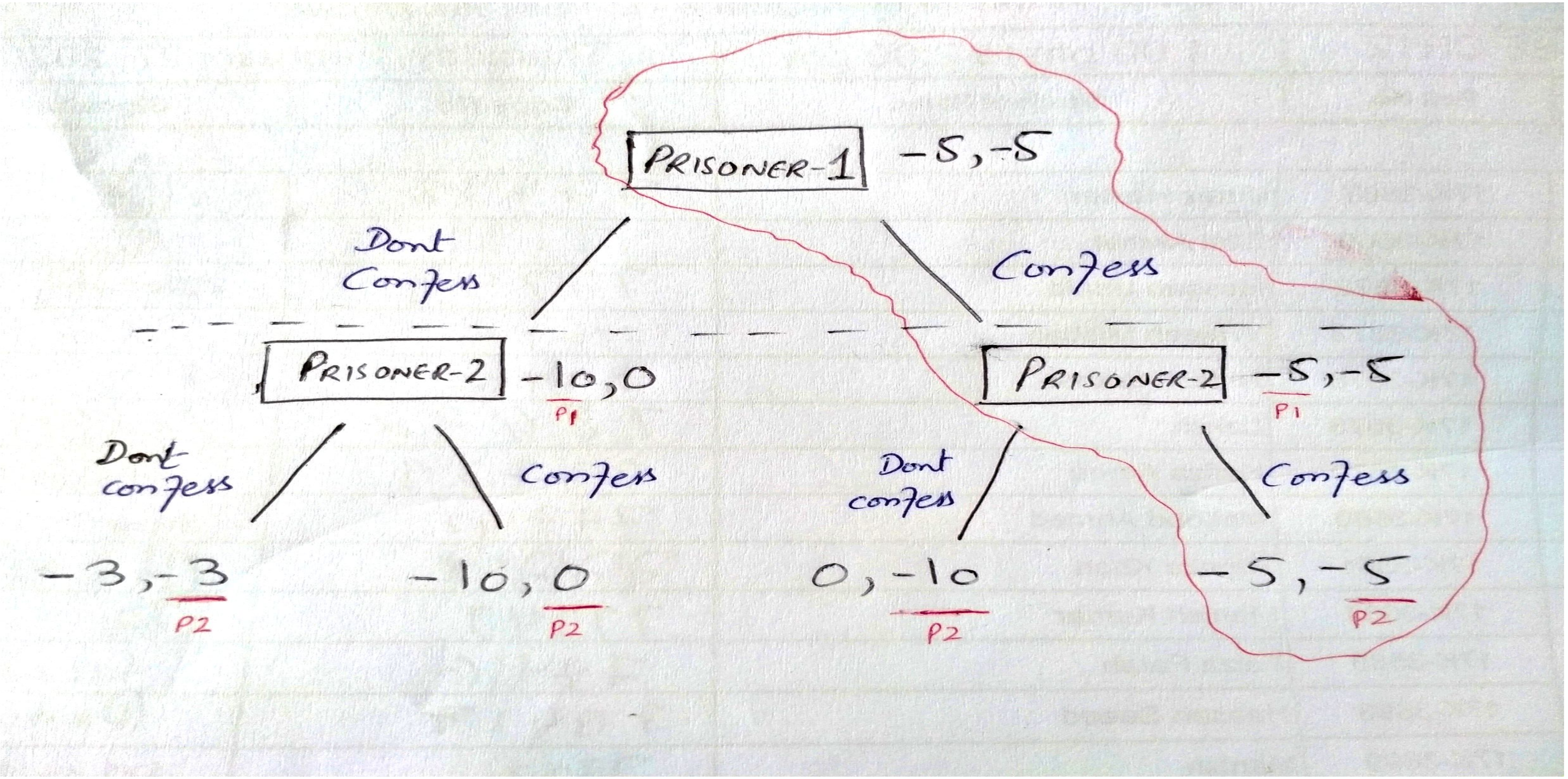
Prisoner’s dilemma

- Two criminals have been arrested and the police visit them separately
- If one player testifies against the other and the other refuses, the one who testified goes free and the one who refused gets a 10-year sentence
- If both players testify against each other, they each get a 5-year sentence
- If both refuse to testify, they each get a **3** -year sentence

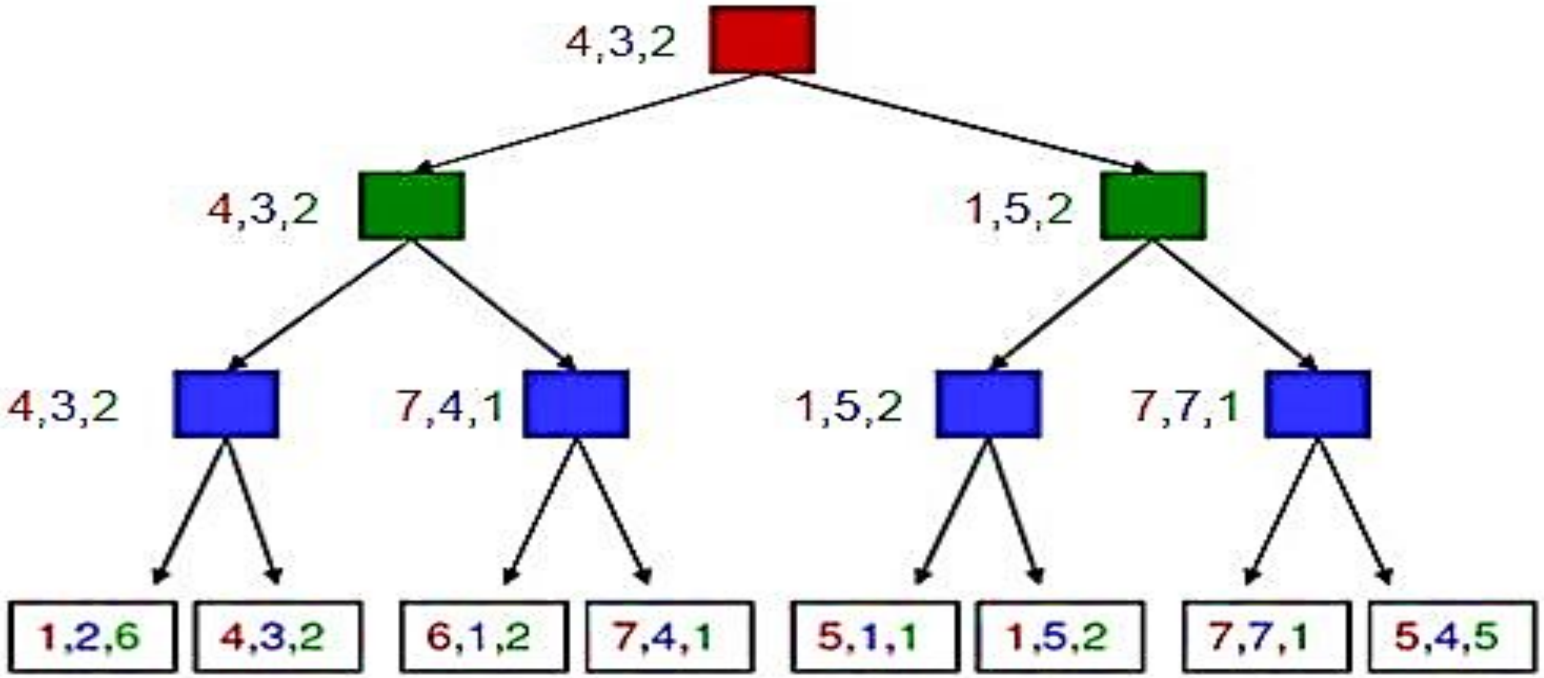
Prisoners’ Dilemma: payoff matrix

1 \ 2	Confess	Don’t Confess
Confess	-5, -5	0, -10
Don’t Confess	-10, 0	-3, -3

Prisoner’s Dilemma in Extensive Form



Multi-player, non-zero-sum games



- Utilities are tuples
- Each player maximizes their own utility at each node
- Utilities get propagated (*backed up*) from children to parents

Properties of Mini-Max algorithm:

- **Complete-** Min-Max algorithm is Complete. It will definitely find a solution (if exist), in the finite search tree.
- **Optimal-** Min-Max algorithm is optimal if both opponents are playing optimally.
- **Time complexity-** As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is $\mathbf{O(b^d)}$, where b is branching factor of the game-tree, and m is the maximum depth of the tree.
- **Space Complexity-** Space complexity of Mini-max algorithm is also similar to DFS which is $\mathbf{O\ (b^d)}$,

Alpha Beta Pruning

- It is an optimization technique for the min-max algorithm.
- In minmax search, algorithm has to traverse all of the possible game states in depth of the tree i.e. **O (b^d)** to make decision.
- **α-β** cuts off branches in the game tree which need not be searched because there already exists a better move available.

The two-parameter can be defined as:

- α= the value of the best (i.e., highest-value) choice we have found so far at any choice point along the path for MAX.
- β = the value of the best (i.e., lowest-value) choice we have found so far at any choice point along the path for MIN.

Alpha Beta Pruning example:

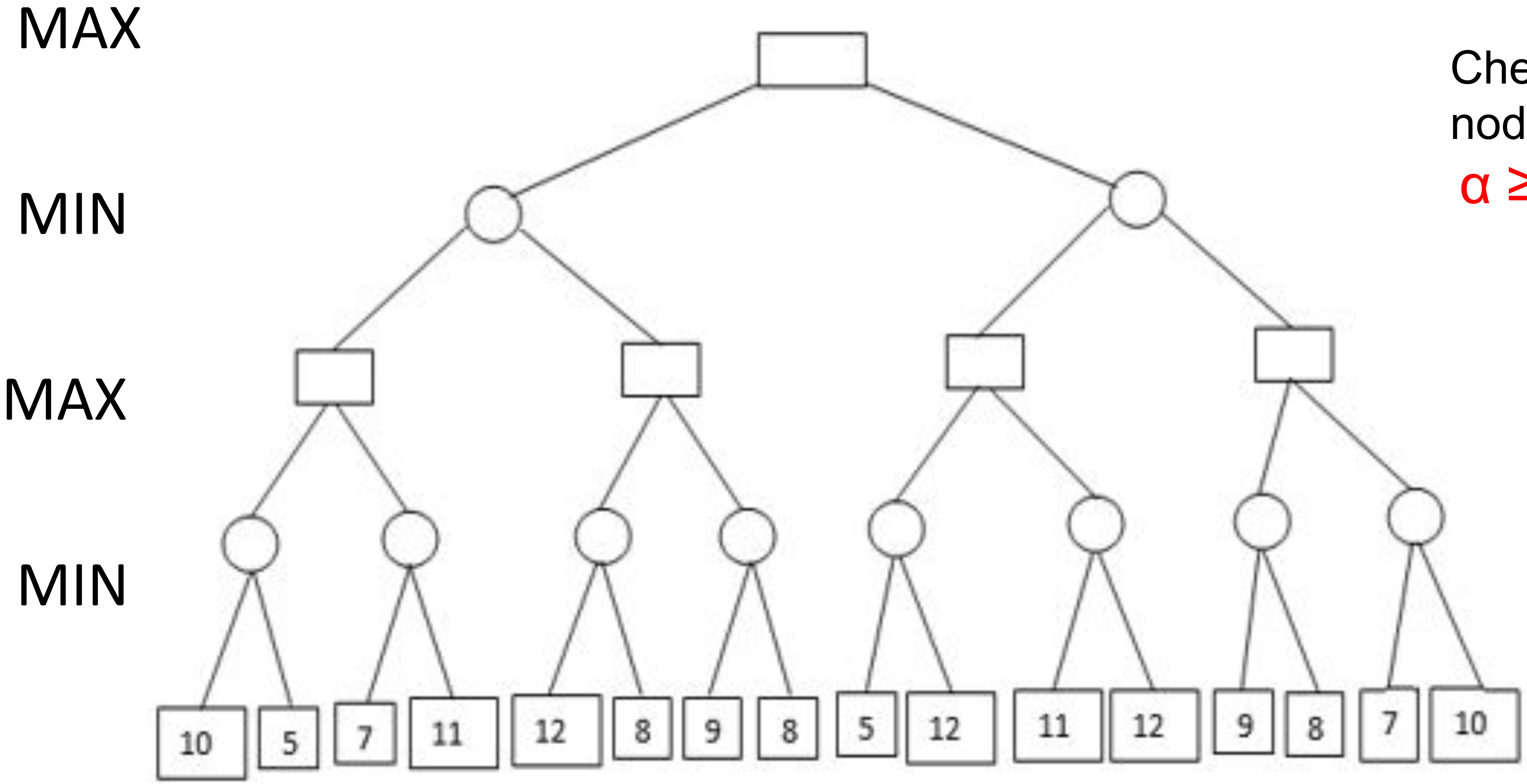
- Considering first node as max.

At each node initially

$\alpha = -\infty \quad \beta = \infty$

Check at each node

$\alpha \geq \beta$(Cut-off)



For MAX Nodes, V=max(α) & α >= -∞

For MIN Nodes, V=min(β) & β >= ∞