



**Faculty of Computing and Information Technology**

**University of the Punjab,  
Lahore**

**Artificial Intelligence Lab 6**

**Instructor: Qamar U Zaman**

# 1. Alpha-Beta Pruning

## Introduction to Alpha-Beta Pruning

Alpha-Beta Pruning is an optimization technique for the Minimax algorithm that significantly reduces the number of nodes evaluated. It achieves this by "pruning" branches in the game tree that won't affect the final decision.

- **Alpha:** The best value that the maximizer can guarantee up to that point.
- **Beta:** The best value that the minimizer can guarantee up to that point.

This technique is typically used in two-player, zero-sum games like Tic-Tac-Toe or Chess, where players alternate turns, and the game outcome is either a win, loss, or draw.

---

## Problem: The Game of Tic-Tac-Toe

In this lab, we will apply Alpha-Beta Pruning to optimize the Minimax algorithm for a game of Tic-Tac-Toe. The objective is to make the AI play optimally while reducing the number of nodes it evaluates to make its move.

---

## Code Template:

```
class AlphaBetaPruning:
    def __init__(self, depth, game_state, player):
        # Initialize the depth, current game state, and player (maximizer or
        minimizer)
        pass

    def is_terminal(self, state):
        # Check if the game has reached a terminal state (win, lose, draw)
        pass

    def utility(self, state):
        # Return the utility value of the terminal state
        pass

    def alphabeta(self, state, depth, alpha, beta, maximizing_player):
        # Implement Alpha-Beta pruning
        pass

    def best_move(self, state):
        # Determine the best move using Alpha-Beta pruning
        pass
```

---

## Lab Tasks:

1. **Implement the Alpha-Beta Pruning algorithm** for the game of Tic-Tac-Toe.
2. Test the algorithm by playing against the AI at different difficulty levels (adjusting depth).
3. **Measure the performance** of the Alpha-Beta Pruning algorithm by counting the number of nodes evaluated for each move and compare it with the basic Minimax algorithm.