Name - Piyusha B. Supe.
Roll No. 23CO315.

AI Activity.

Activity.

- Objective: To apply game theory and alpha beta pruning to a real world game (tic tac toe) by building a game tree and analyzing strategy using evaluation functions.

- Alpha Beta Pruning – It is a modified version of the minimax algorithm. It is an optimization technique for minimax. In minimax the number of game states it has to examine are exponential in depth of tree. Alpha beta pruning can be applied at any depth of the tree.

1) Alpha – The best highest value choice we have found so far at any point along the path of maximizer. The initial value of alpha is $-\infty$.

2) Beta – The best lowest value choice we have found so far at any point along the path of minimizer. The initial value of beta is $+\infty$.

The alpha beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does. but it removes all the nodes which are not really affecting final decision but making algorithm slow.

Algorithm:

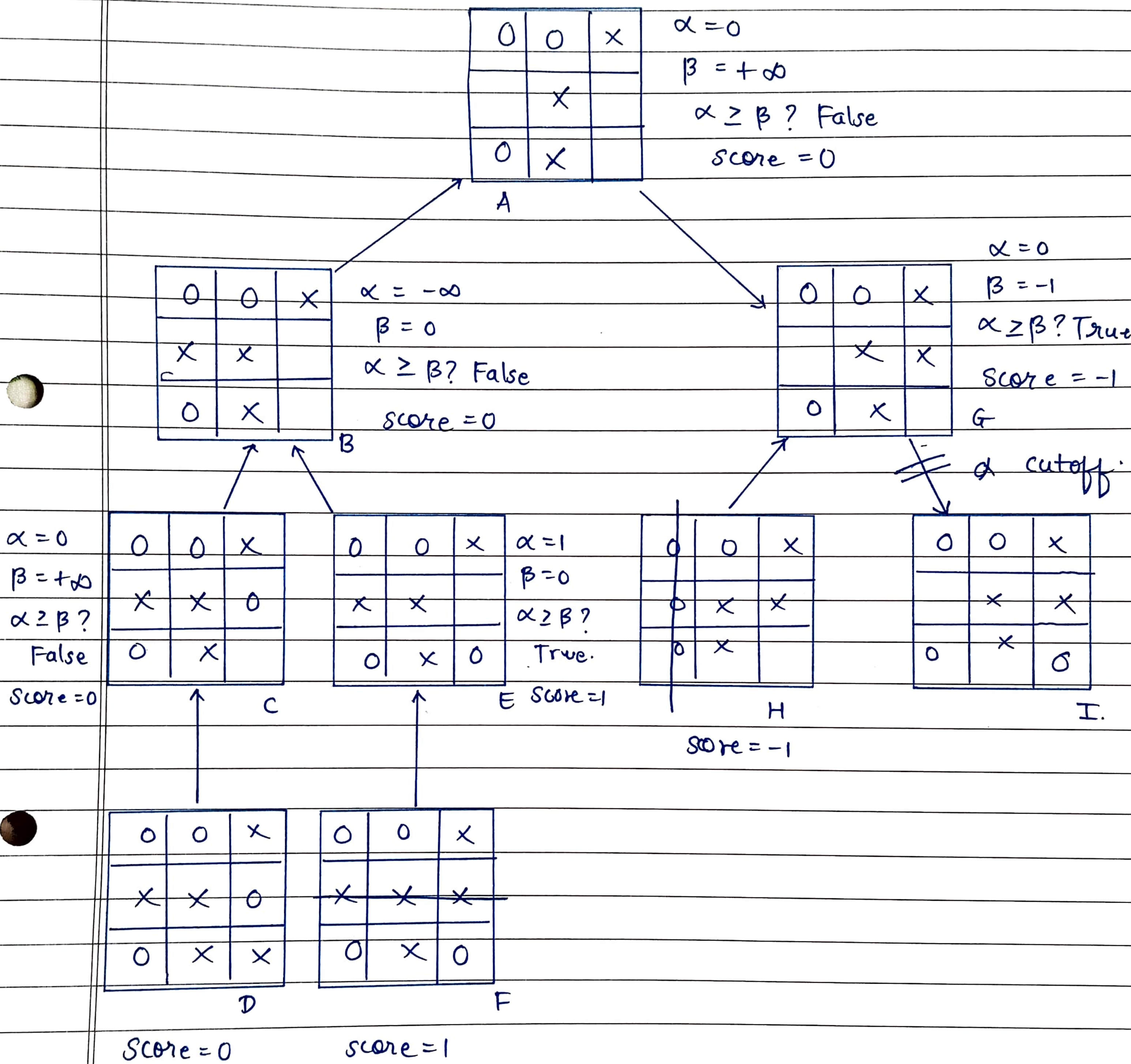
1. First generate the entire game tree starting with the current position of the game all the way up to the terminal states.

2. Apply the utility function to get utility values for all the terminal states.
3. Determine utilities of higher nodes with the help of utilities of terminal nodes.
4. Calculate utility values with the help of leaves considering one layer at a time until the root of the tree.
5. Eventually all backed up values reach to the root of tree. ie. topmost point. At that point MAX has to choose highest value.

- The diagramatic implementation is as follows –

Maximizer = x
choose max among leaf nodes.

Minimizer = 0.
Choose a min among leaf scores.



$\alpha = \max(\alpha, \text{best score})$ $\beta = \min(\beta, \text{best score})$.

Note: In Alpha beta pruning the child node did not visit all the terminal leaf nodes. and its best score may not necessarily be the global best score. (unless its -1 for minimizer and +1 for maximizer).

* Conclusion and reflection –

- Minimax helped choose the best move by anticipating opponent responses.
- Evaluation function gave numeric value to each terminal / near terminal state.
- Alpha beta pruning avoids evaluating non optimal paths saving time and space.
- Optimal move: Node B or C (both give heuristic 0)
  Node A is risky.
- Pruning benefit increases with more complex boards or games like chess.