Report: Othello Assignment

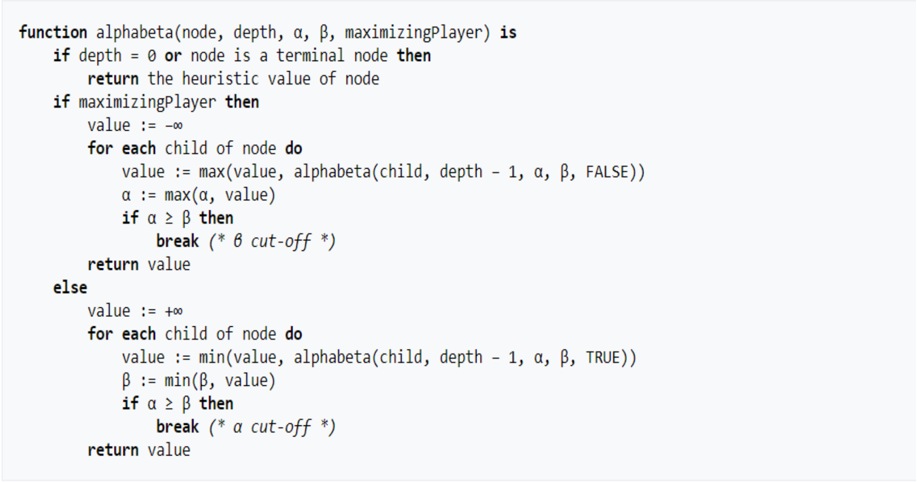
CS18M041 : Pranav Murali
CS18M058 : Vedansh Gurunathan

31 March 2019

The Algorithm

Alpha–beta pruning is a search algorithm that seeks to decrease the number of nodes that are evaluated by the minimax algorithm in its search tree. It is an adversarial search algorithm used commonly for machine playing of two-player games. It stops evaluating a move when at least one possibility has been found that proves the move to be worse than a previously examined move.Alpha-beta pruning is an optimization technique for minimax algorithm. It reduces the computation time by a huge factor. This allows us to search much faster and even go into deeper levels in the game tree. It cuts off branches in the game tree which need not be searched because there already exists a better move available

Below is the pseudo code for alpha beta pruning algorithm



```
function alphabeta(node, depth, α, β, maximizingPlayer) is
    if depth = 0 or node is a terminal node then
        return the heuristic value of node
    if maximizingPlayer then
        value := −∞
        for each child of node do
            value := max(value, alphabeta(child, depth − 1, α, β, FALSE))
            α := max(α, value)
            if α ≥ β then
                break (* β cut-off *)
        return value
    else
        value := +∞
        for each child of node do
            value := min(value, alphabeta(child, depth − 1, α, β, TRUE))
            β := min(β, value)
            if α ≥ β then
                break (* α cut-off *)
        return value
```

Heuristics

We have used the following heuristics:-

1. **Corner Heuristic:-**
   We consider corner squares as the squares present on top-most row, bottom-most row, left-most row and right-most row. The specialty of these squares is that once they are captured, they cannot be flanked by the opponent and allow the player to build the coins around them. Capturing these corners would ensure stability in the region, and stability determines the outcome of the game to a large extent.

2. **Mobility:-**
   One of the strategy to win the game is to restrict your opponents mobility and mobilize yourself.This ensures that the number of potential moves that your opponent has would drastically decrease, and your opponent would not get the opportunity to place coins that might allow him/her to gain control. Mobilizing yourself helps in choosing a move from a larger set of moves which can help in control the proceeding of the game.

   - **Actual Mobility:-**
     Actual Mobility is the number of valid moves you have given the current position.
   - **Potential Mobility:-**
     Potential Mobility gives the potential of the score after a move. It checks the neighbours and if we have neighbour of opposite coin then its a potential switch, so we add the number of opposite neighbours. Similarly for the opponent also we calculate the potential and subtract it.

3. **Weight Heuristic:-**
   We have static board of weights associated with each coin position. The static board captures the importance of each square on the board. The heuristic value for a player is calculated by summing up all the values of the board where his coin is present.

   Below is the static table of weights:-

40	-30	21	8	8	21	-30	40
-10	-17	-4	1	1	-4	-17	-10
21	-4	2	2	2	2	-4	21
8	1	2	1	1	2	1	8
8	1	2	1	1	2	1	8
21	-4	2	2	2	2	-4	21
-10	-17	-4	1	1	-4	-17	-10
40	30	21	8	8	21	-30	40

Final heuristic value for a player is the weighted sum of different heuristic values. We have assigned weights for different heuristics.
Weights are given below:-
Corner heuristic:-2
Actual Mobility:-10
Potential Mobility:-5
Static weight heuristic:-3

References

An Analysis of Heuristics in Othello by Vaishnavi Sannidhanam and Muthukaruppan Annamalai
https://courses.cs.washington.edu/courses/cse573/04au/Project/mini1/RUSSIA/Final$_{P}aper.pdf$