2020

ARTIFICIAL INTELLIGENCE

ASSIGNMENT THREE

NGOMBA LITOMBE

Contents

PROBLEM DEFINITION

Develop an application in a programming language of your choice to play the Reversi game (https://en.wikipedia.org/wiki/Reversi) against a human player, by implementing the alpha-beta method.

Reversi is a strategy board game for two players, played on an 8×8 uncheckered board. There are sixty-four identical game pieces called disks (often spelled "discs"), which are light on one side and dark on the other. Players take turns placing disks on the board with their assigned color facing up. During a play, any disks of the opponent's color that are in a straight line and bounded by the disk just placed and another disk of the current player's color are turned over to the current player's color.

The object of the game is to have the majority of disks turned to display your color when the last playable empty square is filled.

INFORMATION ABOUT APPLICATION

ALGORITHM

The following algorithm is used but in the case of the utility function, an evaluation function is used at the specified depth as measure of how desirable a particular move is.

function ALPHA-BETA-DECISION($state$) **returns** an action
  return the a in ACTIONS($state$) maximizing MIN-VALUE(RESULT(a, $state$))

---

function MAX-VALUE($state, \alpha, \beta$) **returns** *a utility value*
  inputs: $state$, current state in game
    α, the value of the best alternative for MAX along the path to $state$
    β, the value of the best alternative for MIN along the path to $state$
  if TERMINAL-TEST($state$) **then return** UTILITY($state$)
  $v \leftarrow -\infty$
  for a, s in SUCCESSORS($state$) **do**
    $v \leftarrow$ MAX(v, MIN-VALUE(s, α, β))
    if $v \geq \beta$ **then return** v
    $\alpha \leftarrow$ MAX(α, v)
  return v

---

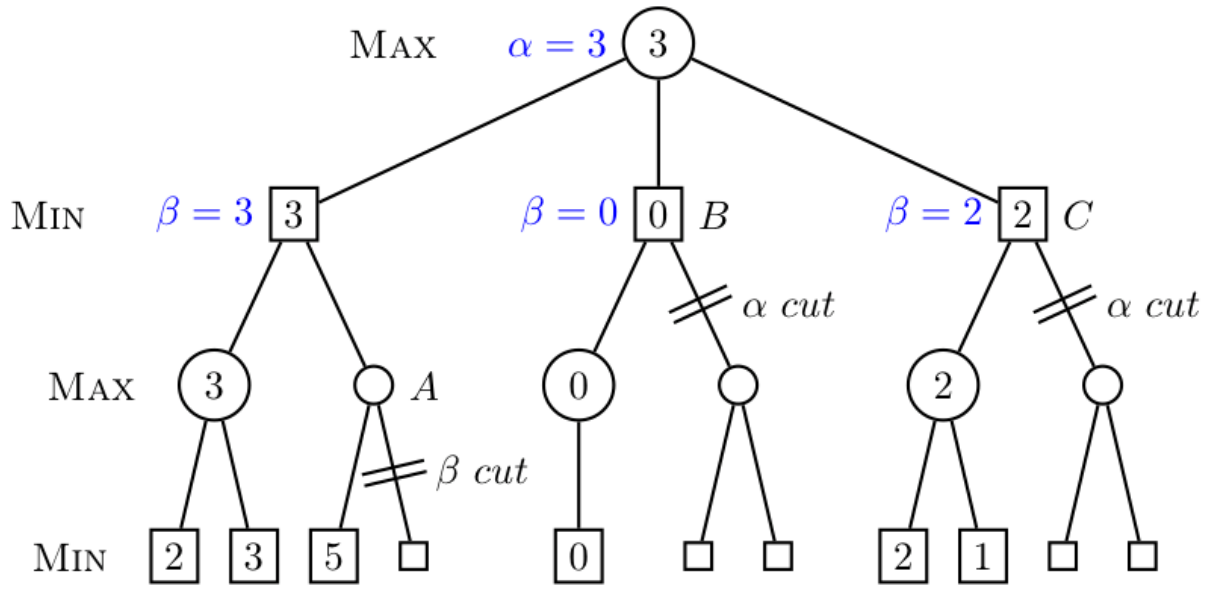
function MIN-VALUE($state, \alpha, \beta$) **returns** *a utility value*
  same as MAX-VALUE but with roles of α, β reversed

Alpha beta is an addition to the min-max algorithm that prevents the exploration of particular leaves in the tree that do not add more information concerning the desirability of a particular play by performing cuts in the tree (alpha and beta cuts).

In min max , the aim of the max player is to maximize his chances of winning the game by selecting the branch of the tree that returns the maximum value (from the eval function at some depth). The aim of the

min player is to reduces the max player's chances of winning the game by choosing the branch with the lowest value.

In the alpha beta better algorithm, we limit the number of visited leaves by doing cuts that would not affect the results. For instance, consider the following example.



The values 2 and 3 are propagated from the last min level to the max play. The max player who wants to maximize his chances of winning selects the play with the highest value, that is 3, and this value is propagated to the higher min play (beta = 3), now at point A, the first branch results in a value of 5. Now since A is a max play, it means the max player would always select a play equal to or greater than 5 at this stage , but 5>3 and the min play above would never select any value >3, therefore there is no need to explore the subsequent branches any further so a beta cut is done.

The value 3 is propagated to the max root node. At B, the first branch gives a 0. Since B is a min play, the chosen value would always be <=0. But the max play above would never select any branch with value < 3. There since 0<3, the subsequent branches of B are cut off (alpha cut).

Pruning does not affect the final results and the behavior of the algorithm is preserved under any monotonic transformation of the eval function. The performance of the algorithm is improved by perfect ordering and can double the solvable depth.

EVAL FUNCTION

A simple evaluation function would be to count the number of white vs the number of black seeds on the board. But this wouldn't be a very wise idea as there are some cells on the board that are more advantageous than others. This means simple counting is not sufficient.

The corner positions for instance are more important than the central positions and the side positions as a seed placed in any of the 4 corners is less likely to be swapped and also poses a higher threat to the opponent. For this reason, I have associated weights to different positions.

- The corners have a weight of 4.
- The side positions have a weight of 2.
- The central positions have weights of 1.

The maximum return from the eval function is (where n is the size of the board):

Max eval(board) = **4**x4 + $(n-2)^2$ + $(n-2)$x4x**2** **(**weights in bold)

= n^2 + 4n + 4

Min eval (board) = 0 (no seed present on the board for that player)

At beginning of the alpha beta algorithm, all the min nodes are initialized with positive infinity and the max nodes are initialized with negative infinity. in my case, I have chosen

Alpha_init = max eval(board) + 1

= n^2 + 4n + 4 + 1 and

Beta_int = min eval(board) – 1

= 0 - 1

DEFINITION OF PROGRAM INPUT PARAMETERS

Size

This is the size of the board. Since we only have 26 alphabets, I have limited this size to 26.

Level

This is depth at which the eval function should be utilized and how far the alpha beta algorithm should go. The higher the level , the more difficult the game for the human player.

Color

Can either be black or white. The black player always starts first.

SAMPLE RUN

```
REVERSI BOARD GAME
1: New game.
2 :Quit game.
Select choice: 1
Enter board size, level(DEFAULT: 4) and colour(white/black): 4 4 white
    a   b   c   d
  +---+---+---+---+
1 |   |   |   |   |
  +---+---+---+---+
2 |   | W | B |   |
  +---+---+---+---+
3 |   | B | W |   |
  +---+---+---+---+
4 |   |   |   |   |
  +---+---+---+---+

Score black: 2
Score white  : 2
```

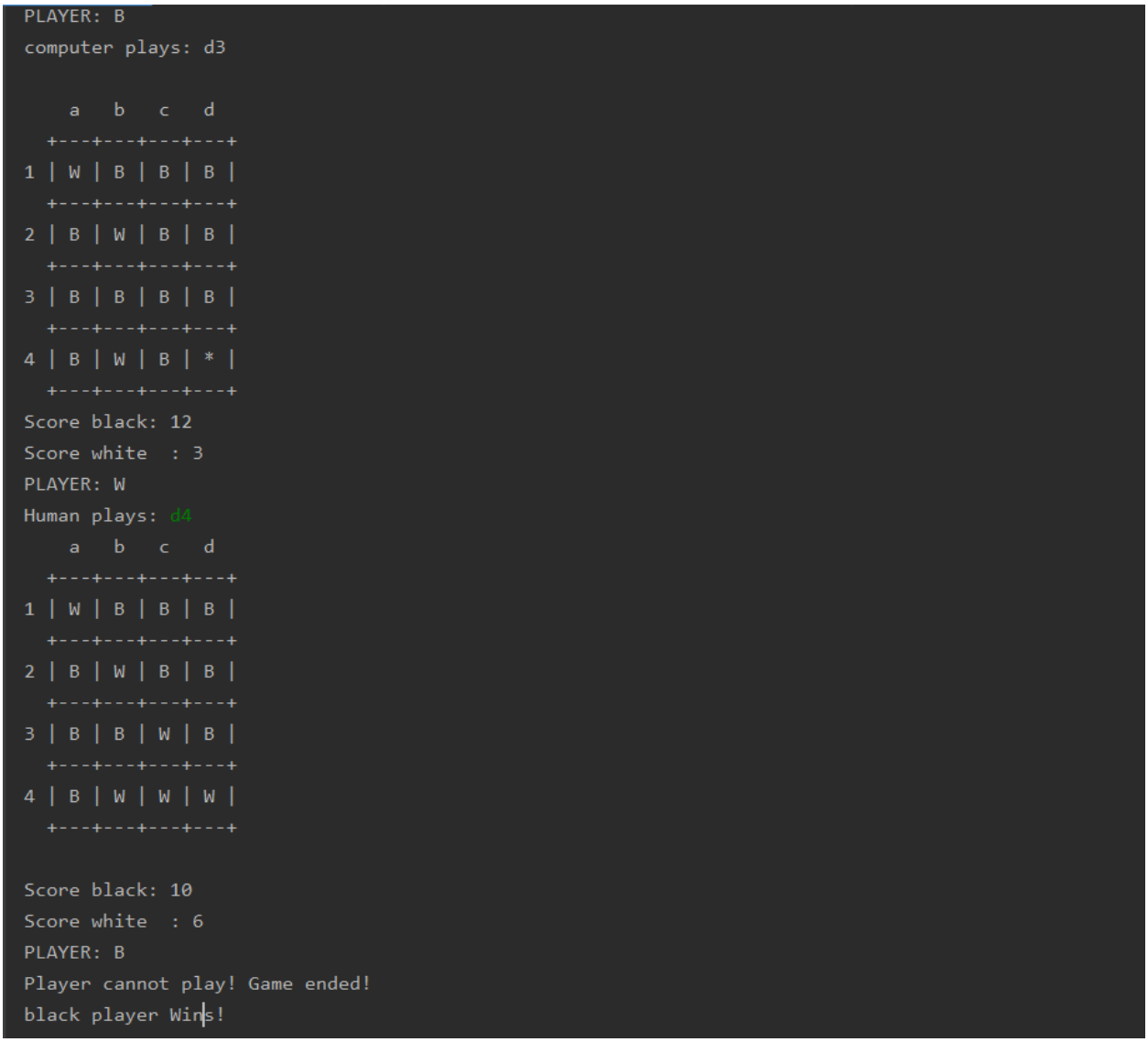
```
PLAYER: B
computer plays: a2

    a   b   c   d
  +---+---+---+---+
1 | * |   | * |   |
  +---+---+---+---+
2 | B | B | B |   |
  +---+---+---+---+
3 | * | B | W |   |
  +---+---+---+---+
4 |   |   |   |   |
  +---+---+---+---+
Score black: 4
Score white  : 1
PLAYER: W
Human plays: c1
    a   b   c   d
  +---+---+---+---+
1 |   |   | W |   |
  +---+---+---+---+
2 | B | B | W |   |
  +---+---+---+---+
3 |   | B | W |   |
  +---+---+---+---+
4 |   |   |   |   |
  +---+---+---+---+

Score black: 3
Score white  : 3
```

```
PLAYER: B
computer plays: d1

    a   b   c   d
  +---+---+---+---+
1 | * |   | W | B |
  +---+---+---+---+
2 | B | B | B |   |
  +---+---+---+---+
3 | * | B | W |   |
  +---+---+---+---+
4 |   |   |   |   |
  +---+---+---+---+
Score black: 5
Score white  : 2
PLAYER: W
Human plays: a3
    a   b   c   d
  +---+---+---+---+
1 |   |   | W | B |
  +---+---+---+---+
2 | B | W | B |   |
  +---+---+---+---+
3 | W | W | W |   |
  +---+---+---+---+
4 |   |   |   |   |
  +---+---+---+---+

Score black: 3
Score white  : 5
```

```
PLAYER: B
computer plays: a4

    a   b   c   d
  +---+---+---+---+
1 |   |   | W | B |
  +---+---+---+---+
2 | B | W | B | * |
  +---+---+---+---+
3 | B | B | W |   |
  +---+---+---+---+
4 | B | * |   |   |
  +---+---+---+---+
Score black: 6
Score white  : 3
PLAYER: W
Human plays: d2
    a   b   c   d
  +---+---+---+---+
1 |   |   | W | B |
  +---+---+---+---+
2 | B | W | W | W |
  +---+---+---+---+
3 | B | B | W |   |
  +---+---+---+---+
4 | B |   |   |   |
  +---+---+---+---+

Score black: 5
Score white  : 5
```

```
PLAYER: B
computer plays: b1

    a   b   c   d
  +---+---+---+---+
1 | * | B | B | B |
  +---+---+---+---+
2 | B | B | W | W |
  +---+---+---+---+
3 | B | B | W |   |
  +---+---+---+---+
4 | B |   |   |   |
  +---+---+---+---+
Score black: 8
Score white  : 3
PLAYER: W
Human plays: a1
    a   b   c   d
  +---+---+---+---+
1 | W | B | B | B |
  +---+---+---+---+
2 | B | W | W | W |
  +---+---+---+---+
3 | B | B | W |   |
  +---+---+---+---+
4 | B |   |   |   |
  +---+---+---+---+

Score black: 7
Score white  : 5
```

```
PLAYER: B
computer plays: c4

    a   b   c   d
  +---+---+---+---+
1 | W | B | B | B |
  +---+---+---+---+
2 | B | W | B | W |
  +---+---+---+---+
3 | B | B | B |   |
  +---+---+---+---+
4 | B | * | B | * |
  +---+---+---+---+
Score black: 10
Score white  : 3
PLAYER: W
Human plays: b4
    a   b   c   d
  +---+---+---+---+
1 | W | B | B | B |
  +---+---+---+---+
2 | B | W | B | W |
  +---+---+---+---+
3 | B | W | W |   |
  +---+---+---+---+
4 | B | W | B |   |
  +---+---+---+---+

Score black: 8
Score white  : 6
```



TERMINATION

The game ends when both players have no valid moves left.