Name: Pratik Jasani

Course: EN 605.645

**Adversarial Search Self-Check**

**Problem 1. Tic-Tac-Toe**

1. **Think up a heuristic for Tic-Tac-Toe. Remember, a heuristic is a function that takes a state of the game and returns a value indicating how good that game state is. It is used in the following way. If I have 4 moves available to me, I can construct a game state for each of those moves “as if” I had made them and calculate the score. I should pick the move with the highest score (and by “I”, I mean the Agent)**

* I would choose a heuristic function that would cover 3 cases. Assuming player O is opponent.
* **Case 1:** Check if player X has won. (Choose max score)
* **Case 2:** Check if player O has won. (Choose min score)
* **Case 3:** There are total 8 checks that tells us if a player has won or not. We can also use these 8 checks to determine how many possibilities player X and player O has to win. To calculate the final score, we calculate possibilities for player X to win – possibilities for player O to win.

1. **Using whatever symmetry shortcuts that are available, use your heuristic function to examine a game tree (extensive form game) from the starting board of Tic-Tac-Toe that expands 3 ply and find the best first move. The main challenge you will find when drawing these trees is that they spread out quickly since asymmetric moves break the subsequent ability to exploit symmetry. Do your best. You may need to turn the paper sideways, write small, think of alternative layouts. You may need to draw it out several times and then redraw a final draft.**

In the below game tree I assume the first player to be 0 so my huristic was calculated by taking possibilities of O’s win – possibilities of x’s win.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  | **o** |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **o** |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **o** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
| **o** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | **o** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | **o** |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **o** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **o** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
| **x x** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  | **x** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  | **x** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
| **x** |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  | **x** |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  |  | **x** |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  |  |
|  |  | **o** |

|  |  |  |
| --- | --- | --- |
| **o** | **x** | **o** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
| **o o** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  | **o** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  | **o** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  |  |
| **o** |  |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  |  |
|  | **o** |  |

6-3 = 3 5-3 = 2 5-2 = 3 5-3 = 2 6-3 = 3 6-3 = 3 6-3 = 3

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **o** |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **o** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
| **o** |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  | **o** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | **o** |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **o** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **o** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  | **o** |  |

|  |  |  |
| --- | --- | --- |
| **o** | **x** |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
| **x x** |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  | **x** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  | **x** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
| **x** |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  | **x** |  |

|  |  |  |
| --- | --- | --- |
| **o** |  |  |
|  |  |  |
|  |  | **x** |

|  |  |  |
| --- | --- | --- |
| **o** | **o** | **x** |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  | **o** |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  |  |
|  |  | **o** |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  | **o** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  |  |
| **o** |  |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
|  |  |  |
|  | **o** |  |

|  |  |  |
| --- | --- | --- |
| **o** |  | **x** |
| **kk o** |  |  |
|  |  |  |

5-3 = 2 5-3 = 2 5-2 = 3 5-3 = 2 5-3 = 2 5-3 = 2 5-3 = 2

Following the similar logic shown above we would be able to calculate scores for the rest of the boards as well. After evaluating the rest of the boards the decision should tell us the best move would be at the center of the board.

**Is this the move you learned as a child?**

Yes, I think I followed a very similar logic when I played tic-tac-toe as a kid. I always choose the corner or the center as my first move.

**Problem 2. Normal Form Games (“Strategic” Games)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Bar 2** | | | | | |
|  |  | **$2** | | **$4** | | **$5** | |
| **Bar 1** | **$2** | **10** | **10** | **14** | **12** | **14** | **15** |
| **$4** | **12** | **14** | **20** | **20** | **28** | **15** |
| **$5** | **15** | **14** | **15** | **28** | **25** | **25** |

**Bar 2 comparison:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Bar 2** | | | |
|  |  | **$4** | | **$5** | |
| **Bar 1** | **$2** | **14** | **12** | **14** | **15** |
| **$4** | **20** | **20** | **28** | **15** |
| **$5** | **15** | **28** | **25** | **25** |

We can see that $4 strategy strongly dominates $2. So, we can drop $2.

**Bar 1 comparison:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Bar 2** | | | |
|  |  | **$4** | | **$5** | |
| **Bar 1** | **$4** | **20** | **20** | **28** | **15** |
| **$5** | **15** | **28** | **25** | **25** |

We can observe that $4 strongly dominates $2 so we can drop $2

**Bar 2 comparison:**

We can see that $4 strongly dominates $5. So, we can drop $5.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Bar 2** | |
|  |  | **$4** | |
| **Bar 1** | **$4** | **20** | **20** |
| **$5** | **15** | **28** |

**Bar 1 comparison:**

We can see that $4 strongly dominates strategy $5. So, we can drop $5.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Bar 2** | |
|  |  | **$4** | |
| **Bar 1** | **$4** | **20** | **20** |

That leaves us with the best strategy.

**Would you believe me if I told you this was State Space Search? How would you formulate this problem as State Space Search? Under Weak SEDS, you can have multiple solutions. How would you reformulate State Space Search to return all solutions instead of just the first found?**

I think there is a way to do this using state space search, but I could not think of a way. One of the solutions that I can think of for this would be to keep track of the path from start to the final state. When the final state is the goal, we can return that path and keep searching instead of stopping. As it keeps searching, we can make sure that we are updating the parent node properly so that if the final state is found again, we can reconstruct the path by back tracing again.