Universität Zürich[UZH]

Introduction to Artificial Intelligence
Exercise Sheet 5

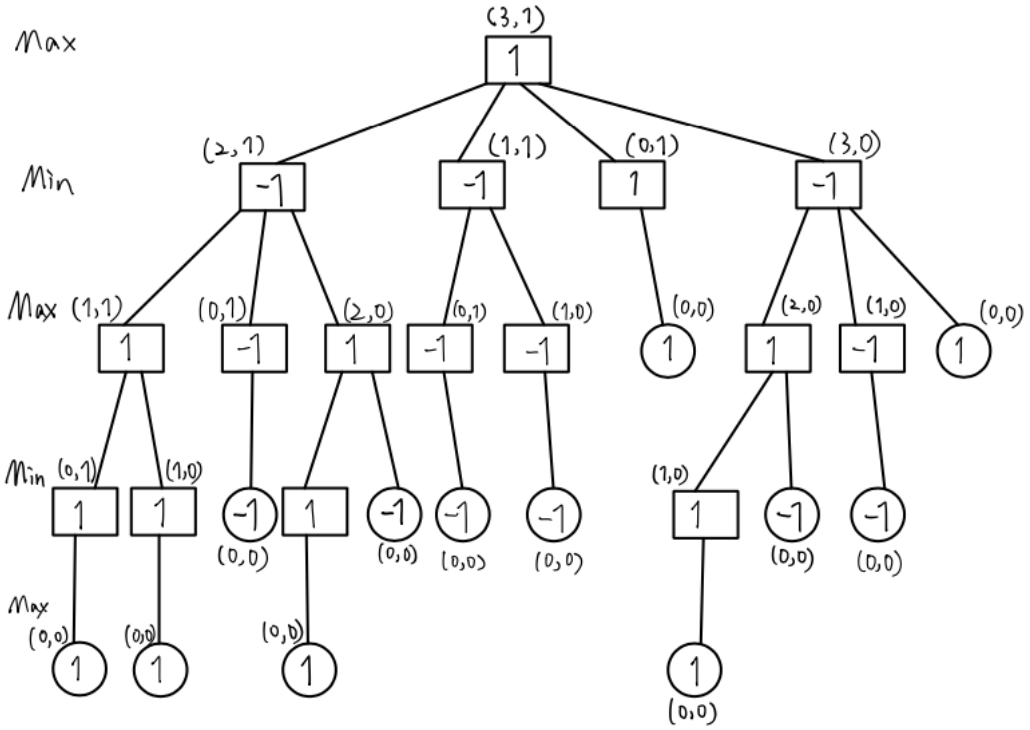
Laurin van den Bergh, 16-744-401
Yufeng Xiao, 19-763-663
Nora Beringer, 19-734-227

Universität Zürich
Institut für Informatik

Due: March 30, 2022

Exercise 5.1

a) Consider the game tree down below:



b) The root value will be 1 as shown in the graph above. Hence MAX has a winning strategy as the value of the root value corresponds to the winning state for MAX which is 1.

The strategy is as follows: If the first player takes exactly two coins out of the stack during the players first move, then the player will win the game.

c) The player who starts has a winning strategy only if the player takes out exactly three coins from Stack 1 during the first move. There is also the possibility to win if Player 1 takes one coin as its first move from Stack 1.

In all other cases the second player wins.

If we picture our two stacks; Stack 1 with 4 coins and Stack 2 with 1 coin and our Player 1 who starts and Player 2.

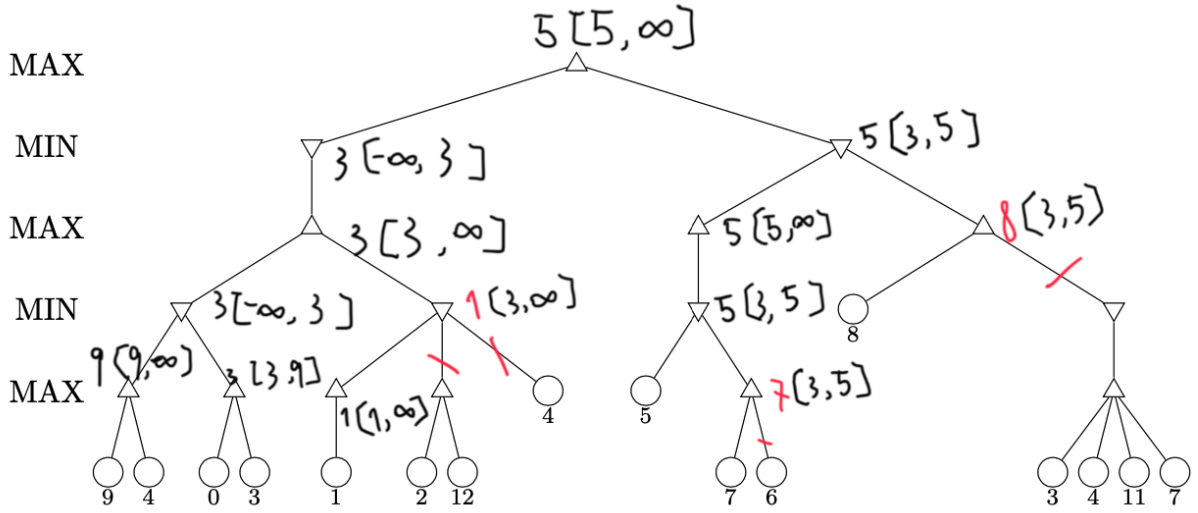
Case 1: Player 1 will take out 3 coins from Stack 1. Now Player 2 can only choose to empty out either one of the stacks fully which in turn leaves the last coin (winning coin) to be removed by Player 1.

Case 2: Player 1 will take out 1 coin Stack 1, then Player 2 can either empty out Stack 2 leaving 3 coins to be taken by Player 1 and in turn winning the game or Player 2 can take 1 coin from Stack 1. Now we have 2 coins on Stack 1 and 1 coin on Stack 2. Player 1 now has to take one coin only from Stack 1 in order to win the game as this would lead

to a Stack 1 with 1 coin and Stack 2 with 1 coin. Now Player 2 can only empty out either one of the Stacks while leaving the last coin (winning coin) to Player 1.
If Player 1 doesn't follow this strategy Player 2 will win the game.
For example: Player 1 takes 1 coin from Stack 1, Player 2 takes 1 coin from Stack 1, Player 1 takes 1 coin from Stack 2, Player 2 can win the game by taking the remaining 3 coins from Stack 1 and so on for all the other game possibilities.

Exercise 5.2

Consider the Tree down below:



Exercise 5.3

1.Iteration: Tree is not yet expanded, hence we don't yet have a MAX or MIN successor to choose. Therefore we start expanding the tree from the root node.
We choose the unexpanded left child and then follow through to the right child by default policy. We reach utility node 16.
We update the value of the lefthand child of the root to 16 and the root to 16.
Tree after 1st Iteration:
2.Iteration: We pick the second child of the root (righthand child) as this child is not expanded yet and therefore we still can't choose the successor for our root node.
We follow by default policy to the right until we reach utility node 14. We update the roots' righthand child to 14. We update the root value as following 16 + (14-16)/2 = 15.
Tree after 2nd Iteration:
3.Iteration: We now can choose the sucessor for our root node as it has two expanded

children. As the root node is MAX we choose the node with the highest utility, which is the node to the left with value 16. From there we have no expanded nodes. Therefore we pick the unexpanded left child and follow to the right by default policy until utility node 6.

We update the value of the lefthand root child to 16 + (6-16)/3 = 38/3 and the root value to 15 +(6-15)/3 = 12.

Tree after 3rd Iteration:

4.Iteration: The root node still has two expanded children but the root and lefthand side were updated with different values, therefore we again have to choose the successor for our root node.

As the root node is MAX we choose the node with the highest utility, which is the node to the right with value 14. From there we have no expanded nodes. We pick the unexpanded left child and follow to the right by default policy until utility node 12.

We update the value of the righthand root child to 14 + (12-14)/4 = 13.5 and the root value to 12 +(12-12)/4 = 12.

Tree after 4th Iteration:

Consider the full Tree below: