

Homework #4

Paolo Grignaffini (ID: 800673343)

May 1, 2018

Set of states

The state are representet through an array of **char** called **conf**, that is a member of the **Node**'s class.

A character equal to ***** means that the point corresponding to the array's index has not been colored/chosen yet, a character equal to **A** means that player A has colored the point, the same applies to character **B**.

Therefore the inital state is represented as an array containing just *****.

Example of possible states: if we take into consideration the graph inserted in the programming assignment paper we could have:

Initial state: $\langle *, *, *, *, * \rangle$

$\langle A, *, *, *, * \rangle \langle A, B, *, *, * \rangle \langle A, B, *, *, A \rangle \dots$

Player function

In order to determine which player has the turn I use the level of the node inside the tree. If the node has an even level, it means that is a node *A* so the player who has the turn is *B*, if the node's level is odd it means that is a *B* node so the player who has the turn is *A*.

Result function

In order to generate the children of a node we need to know who is the player who has the turn and the possible moves that he can make.

If it's *A* turn to play we know that we need to add character *A* in the child configuration, this can be done if and only if the place where we want to insert *A* is occupied by ***** (the algorithm tries them all) and if and only if there are no edges between the possible new *A*'s place and the other *A* characters in the configuration. This is done using an adjacent list that memorize for every node its neighbors, and a vector **AMoves** that contains all the places already occupied by *A*, in this way the result function generates only the node's children that represent a legal configuration. The same applies to *B*.

Termination test

When the result function cannot generate any children for any node in the tree anymore, this means that the game is over, since we have tried all the possible legal configurations. Nodes that do not have children are called leaf and to the is assigned a numeric value by the utility function.

Utility function

The utility function assigns to every leaf a score, if the leaf is an A node then the score is equal to the number of * into the node configuration. if the leaf is a B node then the score is equal to the number of * into the node configuration multiplied by -1. The utility function is called recursively on the root children in order to have a score for every node in the tree, for the nodes who are not leaves the score is determined by picking the max/min of his children.