Up function:

Chart

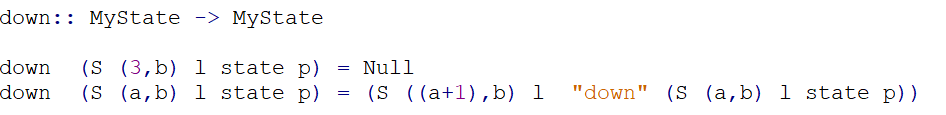
Description automatically generated

The function takes as input a state and returns the state resulting from moving up from the input state by decreasing the x-coordinate of the robot’s location by 1 and changing the state of the robot to “up”. If up results in going out of the boundaries of the 4x4 grid by exceeding the first row (which has index 0), Null is returned.

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state

Down function:



The function takes as input a state and returns the state resulting from moving down from the input state by increasing the x-coordinate of the robot’s location by 1 and changing the state of the robot to “down”. If down results in going out of the boundaries of the 4x4 grid by exceeding the 4th row (which has index 3), Null is returned.

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state

Left function:

Chart

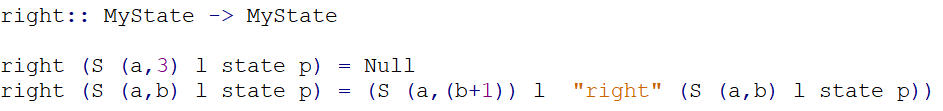
Description automatically generated with medium confidence

The function takes as input a state and returns the state resulting from moving down from the input state by decreasing the y-coordinate of the robot’s location by 1 and changing the state of the robot to “left”. If left results in going out of the boundaries of the 4x4 grid by exceeding the first column (which has index 0), Null is returned.

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state

Right function:



The function takes as input a state and returns the state resulting from moving down from the input state by increasing the y-coordinate of the robot’s location by 1 and changing the state of the robot to “right”. If right results in going out of the boundaries of the 4x4 grid by exceeding the 4th column (which has index 3), Null is returned.

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state

isGoal function:

A picture containing diagram

Description automatically generated

The function takes as input a state and returns true if the input state has no more mines to collect (the list of mines is empty by checking that the length of this list is equal zero using the predefined length function), and false otherwise.

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state

constructSolution function:

Text, letter

Description automatically generated

The function takes as input a state and returns a list of strings representing actions that the robot can follow to reach the input state from the initial state. That is done by checking if the action performed now by the robot is not equal to an empty string (“”) then call the function constructSolution recursively on the parent state (called p) and append to its end a list consisting of the action performed now by the robot (called state) using the predefined append function (++), otherwise return an empty list because now we are at the initial state of the robot. The possible strings in the output list of strings are only "up", "down", "left", "right", and "collect".

Variables:

* a: the x-coordinate of robot’s location
* b: the y-coordinate of robot’s location
* l: the list of mine coordinates
* state: String representing the last action performed to reach this state (up,down,right,left,collect)
* p: parent state