Task 1: PART A – Solving Algorithms

By/ Ismail Farahat, 23/8/2019

1-The Algorithm

The algorithm is to make turtle move in every possible path to find the coins and to find the finish point and we are going to do that by considering all the possibilities that the turtle is going to face in its path…so we are going to build 3 functions by python to check the blocks around the turtle…these functions are:

- **check_left (CL)**
  returns true if the left side of the turtle has a block belong to wall list
- **check_right (CR)**
  returns true if the right side of the turtle has a block belong to wall list
- **check_forward (CF)**
  returns true if the forward path of the turtle has a block belong to wall list

Then we are going to build the truth table to give the turtle 4 orders:

- **rotate left (L)**
- **rotate right (R)**
- **move backward (B)**
- **move forward (F)**

sometimes, the turtle go to end dead in any path and when that happens the turtle will come back but the problem is, it could come back to the same first path and moves in single loop to infinity, so we need to block the coordinates that have intersection between more than one way and that will happen by appending these coordinates to wall…so we are going to make another order called **W** to the truth table

THE TRUTH TABLE

input	input	input	output	output	output	output	output
CL	**CR**	**CF**	**L**	**R**	**F**	**B**	**W**
0	0	0	1	0	0	0	1
0	0	1	1	0	0	0	1
0	1	0	1	0	0	0	1
0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	0	1	0	0	0
1	1	0	0	0	1	0	0
1	1	1	0	0	0	1	0

We are going to use logic expressions to express every output (order to the turtle) …and we are going to use 'if condition' in python to program these expressions

L = not (CL)

R = CL and not (CR)

F = CL and CR and CF

B = CL and CR and not (CF)

W = (not (CR) and not (CL)) or
(not (CR) and CL and not (CF)) or
(not (CL) and CR and not (CF))

There are many other small things like collecting the coins…etc. These things are explained inside the code.