Informatics Institute of Technology

In collaboration with

University of Westminster

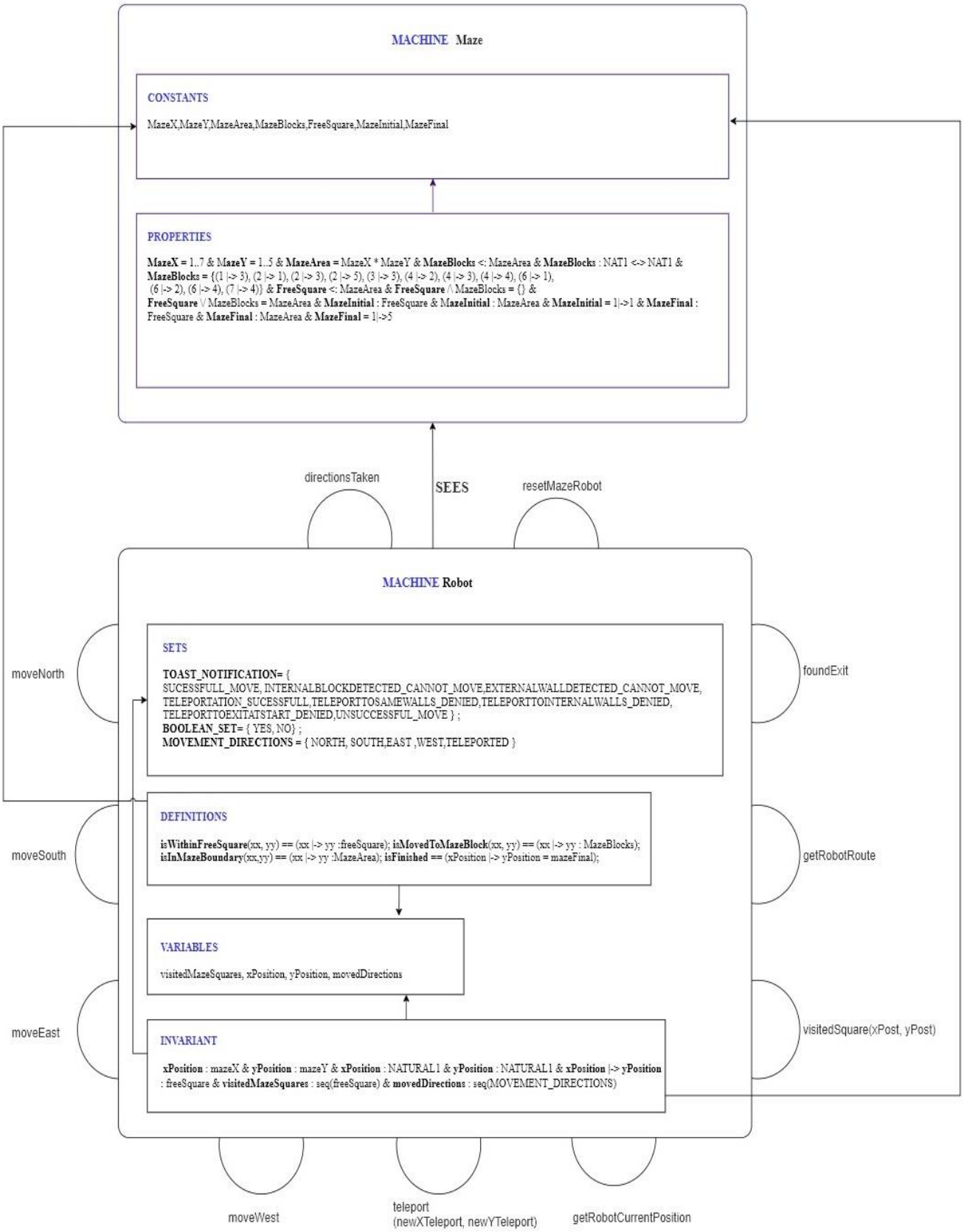
BEng. (Hons) in Software Engineering

2020/2021

Reasoning About Program Coursework

Thivya Thogesan |2017343|w1698503

Structure Diagram



State Invariants explained in plain English

- **xPosition : MazeX**

  The x position of the robot should be within the x-axis boundary of the maze

- **yPosition : MazeY**

  The y position of the robot should be within the y-axis boundary of the maze

- **xPosition** : **NATURAL1**

  The x position of the robot should be in a range of natural number starting from 1

- **yPosition : NATURAL1**

  The y position of the robot should be in a range of natural number starting from 1

- **xPosition |-> yPosition : FreeSquare**

  The current position of the robot should occur or be located in the free square of the known maze.

- **visitedMazeSquares : seq(FreeSquare)**

  The visited path of the robot should be a sequence of free square region.

- **movedDirections : seq(MOVEMENT_DIRECTIONS)**

  The robot's taken movements should be a sequence of its possible SET of DIRECTIONS to move.

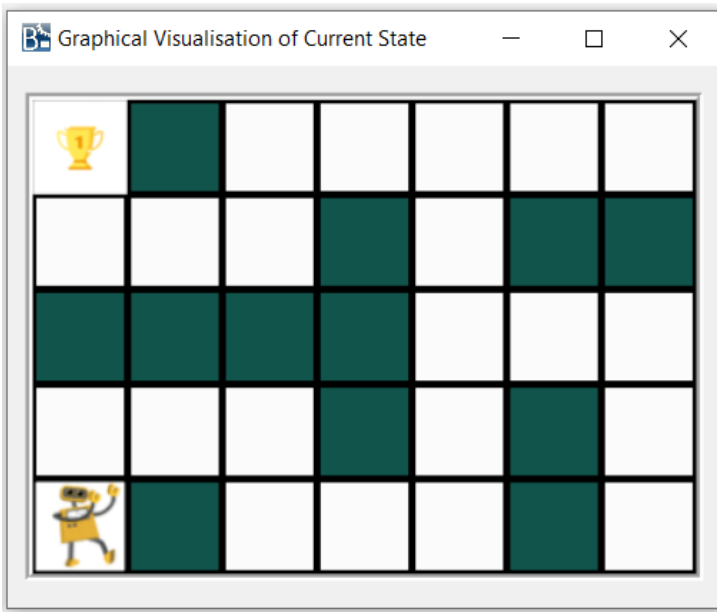
  MOVEMENT_DIRECTION SET consist of North,South,East,West,Teleport.

Note: "free square" refers to an empty cell, that is the cell which is not a blocked cell , in which the robot can locate.

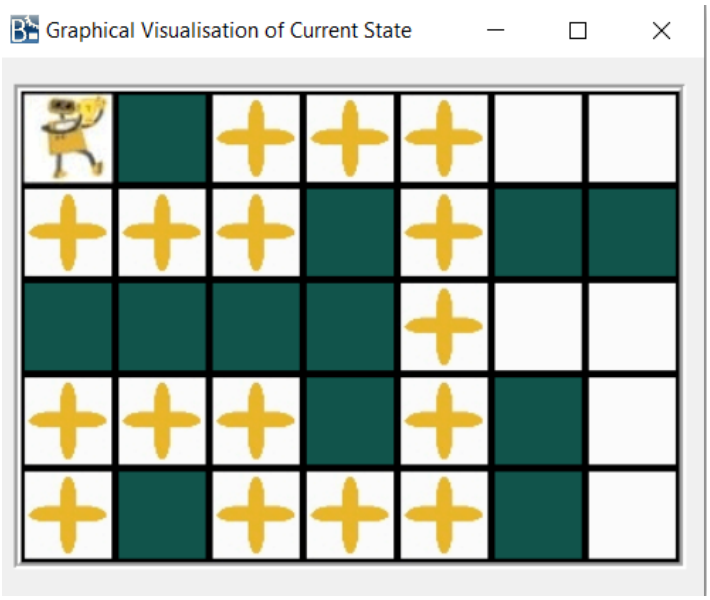
Additional Work – Graphical Representation of a State

As seen below, when animating, the state of the robot is graphically visualized.

An ANIMATION FUNCTION was used to achieve this, the code to achieve rendering of cell each time a state gets updated is shown below.



ANIMATION FUNCTION3 was used to visualize the visited squares by the robot from the initial square of the maze to the final terminal, cross images were used to show the visited square and the final square of the robot was shown using a different image. Combining both these the visited square graphical representation was shown.



Code :

```
 /** ANIMATION FOR ROBOT STATE **/

    ANIMATION_IMG0 == "images/maze.gif";
    ANIMATION_IMG1 == "images/block.gif";
    ANIMATION_IMG2 == "images/robot.gif";
    ANIMATION_IMG3 == "images/winning.gif";
    ANIMATION_IMG4 == "images/spaceship.gif";
    ANIMATION_IMG5 == "images/cross.gif";
    ANIMATION_IMG6 == "images/finalrobot.gif";

   ANIMATION_FUNCTION2 == ({rr,cc,ii |  (cc = xPosition & rr =
(6 - yPosition)) &  isFinished& ii = 6
   });

   ANIMATION_FUNCTION3 == ({rr,cc,ii | cc |-> (6 - rr) :
ran(visitedMazeSquares)&  isFinished & ii = 5
   });


    ANIMATION_FUNCTION_DEFAULT == ({rr,cc,ii | rr:mazeY &
cc:mazeX &
            (IF (cc = prj1(mazeX, mazeY)(mazeInitial) & rr =
prj2(mazeX, mazeY)(mazeInitial))
                THEN
                    // initial square
                    ii = 3
                ELSE
                    IF (cc |-> (6 - rr) : MazeBlocks)
                    THEN
                        // blocked Square

                        ii = 1
                    ELSE
                        IF (cc = xPosition & rr = (6 -
yPosition))
                        THEN
                            // Robot
                            ii = 2
                        ELSE
                            // Free square
                            ii = 0
                        END
                    END

       END)});
```