Assignment 6 – Harshit Mawandia

Q3. Bonus: Maze Traversal:

Our primary function is *traverseMaze,* and we do not use any helper functions.

3. Correctness Proof:

traverseMaze(mazefile):

First, we open the maze file and store all the data for the maze into a list.

Then we make a copy of the maze, removing any embedded spaces between 2 characters.

Then we find the coordinates of the starting position within the maze

Then we initialize the x and y indexes with the starting position

We have 3 invariants within the loop:

- b always remains true while the control is within the loop.
- x and y always store the indexes of the position we are currently at within the maze.
- path stores the path that we have taken to reach those x and y index.

Case 1:

When we move to the next step within the maze, we first check if that is a '_' character or not; if it is, only then we move.

Also, we check if it's possible to move in all four directions one by one. If in any case movement is possible, we adjust the x and y coordinate accordingly, also we change that place's character from '_' to '1' so that we know that we have been there before and do not reach there again and get stuck in an infinite loop.

We also append the direction in which we had moved in the path.

We can see that all the invariants remain satisfied in this case.

Case 2:

If the next character is 'E', we set b to false, append the path with the step we had taken and return the path. Still, all the invariants remain satisfied.

Case 3:

If there is no possible path ahead, we pop the path to remove the last step, adjust the x and y coordinates accordingly, and find another path until we find one containing the exit.