**To:** Dr. Berry  
**From:** Matthew Schack, Peter Heath, and Data  
**Date:** 2/2/16 **RE:** Lab-07 Path Planning

In this lab we implemented topological navigation and path planning as well as metric path planning. For the topological navigation we used the table shown as Table 1 to determine the state the robot is in. Using the state the robot is in and directions in the form of left, right, or straight put in by the user, Data was able to successfully navigate the given path in the world.



**Table 1:** Table used to determine state which Data is in, red represents false and green represents true.

For Metric path planning Data starts with a map of the world on the robot the user then inputs its start and end location. Data then runs a recursive wavefront algorithm to assign each cell a distance from the goal. Data moves along the path from by moving to the cell with the lowest number each time, this allows Data to move the most efficient path.

The recursive wavefront algorithm starts at the goal position setting it to zero. It then runs the same function for every valid point increasing the number it is setting the point in the array to by one. If the function runs on a cell with a value lower than the value it is trying to set the cell to it does not replace the value in the cell. This method allowed Data to tell how far away it was from the goal position at every valid cell. This enabled Data to find the most optimal path by running through the map starting at the start position and only moving in directions that is exactly one less than the cell Data is currently at.

We were able to account for every edge case and as such Data never got stuck at any points.

It took Data about 0-60 seconds to move from a start position to the goal depending on where the start or goal points were.

Implementing an 8 square wavefront would make Data move on a quicker path. However, we then run the risk of having Data get stuck on corners or having error while turning.

Move forward

Wavefront and topological path planning

Input start and goal

Create map

Create path

Read movement

Turn to direction

Topological Navigation

Input directions

Move forward

Is current position a gateway?

Yes

No

Iterate through movement list

Perform inputted movement