AIR - Homework 4

Maximilian Mensing
Torsten Jandt

October 23, 2016

BFS In each iteration, BFS checks if the nodes above, below, left and right of the already searched nodes contain a goal, an obstacle and if they have already been visited. If the checked node has not been visited and contains no obstacle it is marked as visited. The algorithm traverses the search space in all 4 directions at once. If a goal has been found it's position gets logged and marked.

DFS DFS explores the nodes in the search space one by one. It starts by checking the nodes right of the start position until an obstacle is reached. Once this is the case, the exploration continues in the node below and the direction of search gets flipped. If an obstacle is reached in the current horizontal direction and below the current node, the algorithm continues to traverse the unexplored space upwards, starting at the closest unexplored position.

Comparison On *map one* both algorithms store 2323 nodes and visit a total of 1261 nodes. In case of *map two* BFS as well as DFS store 1975 and visit 1145 nodes. On the *third map* BFS and DFS store 4618 and visits 2460 nodes. The Same number of visited nodes on the three results from the fact, that both algorithms check all reachable goals for the presence of a goal. They are not provided with any information on how many goals exist and therefore they can not stop once all goals have been found. The same statement applies for the number of stored nodes. If the number of goals to be found would be provided to the algorithms they would stop traversing the search space once all goal have been found, resulting in differing amounts of stored nodes. In all experiments, the maximum size of the stack / queue equals the number of stored nodes. This is caused by a bug in the provided code.