Programming Assignment 1 í Gervigreind

Tasks:

1) Develop a model of the environment. Define what constitutes a state, successor states, legal actions etc. Consider trying to find abstractions that shrink the state space.

The size of the grid and the locations of the obstacles make up the static portion of the environment. The location of our agent as well as its facing will define which moves are legal and which ones are not and the locations of dirty cells will determine which solution is optimal.

Initial model of the environment: We will use a two dimensional array to represent our grid and employ cartesian coordinates to map the grid to the array. Array values indexed by coordinates mapping to obstacles will be set to 1 and values mapping to dusty cells will be set to 2. All others will be initialized to 0.

A state of the environment is a description of the grid and the obstacles it contains together with information about which cells are dirty, where our agent is currently located and where it is facing.   
A successor state will be the same as its predecessor after updating the agent’s location and facing, taking into account any dirt that may have been removed. I.e after executing TURN.RIGHT the successor state is exactly as the predecessor with the facing updated.

The TURN\_ON move is only legal in the initial state.

TURN\_RIGHT and TURN\_LEFT are legal as long as the state of the agent is ON.

GO is legal if: i) The agent is facing north in cell (X,Y) and cell(X,Y+1) = 0 ii)The agent is facing south in cell(X,Y) and cell(X,Y-1) = 0 iii)The agent is facing west in cell (X,Y) and cell(X-1,Y) = 0 iv) The agent is facing east in cell (X,Y) and cell(X+1,Y) = 0.

SUCK is legal so long as the agent is on.

TURN\_OFF is legal as long as the agent is on.

3)

Each cell can be blank, contain an obstacle or contain dirt. Every cell not containing an obstacle might contain the robot in one of four different facings and two different activation states. Disregarding the obstacles and the states of activation we can distribute the D dirty cells in (W\*L)!/(W\*L-D)! Different ways. For each of those permutations we can choose a location for our robot in W\*L ways and a facing in one of 4 different ways. The size of the state space is therefore approximately ((W\*L)!/(W\*L-D)!)\*WL\*4.

4. Assess the following blind search algorithms: Depth First Search, Breadth First Search, Uniform-Cost Search with regard to optimality, completeness and complexity.

Depth First Search has a time complexity bm where m is the maximum depth of the search tree and b is the branching factor. It has space complexity b\*m. It is neither optimal nor complete in the general case but as this is a finite environment it is complete here. It can’t be made optimal as not all actions have the same associated path cost.  
Breadth First Search has time complexity bd where d is the depth of the first encountered solution as well as space complexity bd. It is a complete search in that it will always find a solution provided there is one to be found. It is however only optimal in the case that the step cost is uniformly 1. Which might not be the case.  
Uniform cost Search is the same as breadth first search for problems where the only path cost is 1 and otherwise it is optimal whereas Breadth First would not be.