n\*n: the number of grids in the parking lot, which can be seen as a board

v: the number of vehicles in the parking lot

1. State space: the set of all reachable and different arrangement for the board. The states are reached by taking a sequence of action described as below. “Reachable” means that at most one occupies one grid.

2. Initial state: the given board

3. Actions:

general case: move one car with a specified car number (e.g. i) one grid from its original grid with a specified direction. The option for direction for one car is either {left, right}, {up, down}.

edge cases: (?) direction

4. Transition Model: returns the board with the specified car (the i-th car) moved by 1 grid to the specified direction (function)

5. Goal test: the red car occupies the cell with the door on one of its edges

6. Lower and upper bound on the branching factor of this formulation, if computable

Lower bound: 0, when there is no valid action (no vehicles can be moved)

Upper bound: 2v, when all cars can be moved to both directions

7. Lower and upper bound on the solution depth of this formulation, if computable

Lower bound: 0, when the red car already occupies the cell with the door on one of its edges

Upper bound: can be compute? Infinitely many when it is not solvable? (A\* don’t go back and forth between states),

8. Optimal solutions to the 3 initial states provided above. For each state, specify the number of moves in the

optimal solution, as well as the sequence of states in the optimal solution. Each state should be represented

textually according to your state representation. There’s no need for fancy graphics, but try to make it so that it is

easy to see where the vehicles are located.