Artificial intelligence programming
Report module 1

Alexander Kjeserud

August 31, 2017

The objective in this assignment is to implement the A* algorithm, and use it to find an optimal solution for the four different difficulties in the game Rush hour.

The heuristic function is the same for all four difficulties. $h(state) = 2^{Board_x - Car0_x}$ Where $Board_x$ is the length of the board, and $Car0_x$ is the x position of the car that is to be moved to the far right, in a given state. The heuristic decreases by the power of two as Car0 approaches the right side of the board.

The node expansion function generates new states by first generating a list of all occupied coordinates in the environment. Then every car in a given state is moved forwards and backwards, to see if the new position is valid. Valid successor states are aggregated and returned.

Performance table

	Puzzle variants															
Search method	Easy-3				Medium-1				Hard-3				Expert-2			
Breath-1st	412,	100,	95,	16	4264,	805,	751,	24	11485,	2041,	1706,	33	83498,	10934,	10143,	73
Depth-1st	202,	109,	47,	42	6944,	1683,	1170,	267	3816,	1825,	551,	473	22127,	10462,	2804,	2486
Best-1st	272,	83,	65,	16	3277,	642,	565,	24	3912,	782,	613,	33	32471,	4819,	4417,	73

Table 1: Comparison of A*, depth-first and breadth-first search on 4 rush-hour puzzles. The four tuples represents total number of states generated, number of unique states in graph, number of node expansions and number of moves from initial state to termination state.