Artificial Intelligence 2019
Problem Sheet 3

Andreas Birk

Notes

The homework serves as preparation for the exams. It is strongly recommended that you solve them before the given deadline - but you do not need to hand them in. Feel free to work on the problems as a group - this is even recommended.

1 Problem

Let $S = s_0, s_1, ...$ denote a set of states and $A = a_0, a_1, ...$ a set of actions where each action a_i gets an agent from one state s_j to an other state s_k, which we denote as $a_i = (s_j, s_k)$. One concrete, very simple example is $S = s_0, s_1$ and $A = a_0$ with $a_0 = (s_0, s_1)$.

Find a combination of states and actions (as simple as possible) such that problem-solving with a BFS tree search runs into an infinite loop. Simply assume that s_0 is the start state and the "last" state in A, i.e., $\max_i s_i$, is the goal state for the problem.

Furthermore, find a combination of states and actions (again, as simple as possible) such that problem-solving with a DFS tree search runs into an infinite loop.

2 Problem

Let $S = s_0, s_1, ..., s_8$ denote a set of states and $A = a_0, a_1, a_2$ a set of actions where each action a_i gets an agent from some state s_j to an other state s_k, which we denote as $a_i = (s_j, s_k)$. Suppose it holds that:

- $\forall i \in \{0, ..., 5\} : a_0 = (s_i, s_{i+3})$
- $\forall i \in \{0, ..., 8\} : a_1 = (s_i, s_0)$
- $\forall i \in \{1, ..., 8\} : a_2 = (s_i, s_{i-1})$

Given the task to find a plan, i.e., a sequence of actions, that gets an agent from s_0 to s_8. Write down the resulting plan and the sequence in which the search space is traversed for a) BFS and b) DFS.

3 (Bonus) Problem

An agent faces three stacks $s_1, ..., s_3$ of toy building blocks. The blocks are colored in **r**ed, **g**reen, and **b**lue. The agent can always only move the top block from one stack s_i and move it to the top of an other stack s_j with the action a_{ij}.

Suppose for each color there is the same number n of blocks. Furthermore, the stacks have equal heights of n blocks in the beginning but they consist of blocks with randomly mixed colors. The task is to sort the blocks by color, i.e., to have all red blocks in s_1, all green ones in s_2, and all blue ones in s_3.

Now, think about the state space of this search problem, i.e., how many different constellations of towers are there? What is the size of the search space for $n = 3$, n in general?

4 (Bonus) Problem

Given the above problem with the colored blocks. Suppose we want to use A^* to solve it. Do we need a admissible or a consistent heuristic? Why?

Formulate (at least) two suited heuristics that could be used in A^* search. Shortly motivate if/why they are admissible, respectively consistent.

5 Problem

Given the following weighted graph with directed edges:

$V = \{v_0, v_1, ..., v_5\}$
$E = \{e_0 = (v_0, v_1), e_1 = (v_0, v_2), e_2 = (v_1, v_2), e_3 = (v_2, v_3), e_4 = (v_2, v_4), e_5 = (v_4, v_5), e_6 = (v_1, v_5)\}$
$W = \{w(e_0) = 1, w(e_1) = 3, w(e_2) = 1, w(e_3) = 2, w(e_4) = 2, w(e_5) = 1, w(e_6) = 5\}$

Suppose Dijkstra is used to search a shortest path from v_0 to all other vertices. Write down the content of the priority queue and the distance estimates for the different vertices in each step of Dijkstra.

6 Problem

Dijkstra (like many other algorithms) uses a priority queue, which can be efficiently implemented using a binary heap.

Suppose the numbers 56, 38, 12, 27, 88, and 92 are inserted into a binary heap (starting with an empty heap). What does the heap in the end look like?

Suppose the minimum is then extracted from the heap. What does the heap look like after this operation?

7 Problem

Given the weight graph $G = (V, E)$ with

- $V = \{v_0, v_1, ..., v_4\}$
- $E = \{e_0 = (v_0, v_1), e_1 = (v_0, v_2), e_2 = (v_0, v_3), e_3 = (v_1, v_4), e_4 = (v_3, v_4)\}$
- $W = \{w(e_0) = 5, w(e_1) = 1, w(e_2) = 3, w(e_3) = 9, w(e_4) = 2\}$

The heuristic function $h()$ is discretely defined per vertex, i.e.,

- $h(v_1) = 4$, $h(v_2) = 2$, $h(v_3) = 3$, $h(v_4) = 0$

Use A* to find the shortest path from v_0 to v_4.