Foundations of Artificial Intelligence
Exercise Sheet 2

Robin Vogt, Riccardo Salvalaggio

September 23, 2021

Exercise 2.1

Rubik's Cube

State space The state space is finite and there are approximately 43 quintillion different states that can be reached by turning sides of the cube. The different states can be described by the colors of the nine tiles on each of the six faces.

Initial state The initial state is just one random permutation of the the tiles on each faces, that can be reached by turning sides of an initially solved cube.

Actions Every Action is a rotation by 90° of a column or row of one of the faces to the right or left, top or down, respectively.

Goal test Check whether all tiles on each of the six faces have the same color.

Path cost function The cost of each action is one unit cost. The path cost function is therefore simply the length of an action sequence.

0.1 Map of Europe

State space Each of the 27 countries is either blank or colored in one of the four possible colors.

Initial State A complete blank map

Actions An action is to color exactly one still blank country in one of the four possible colors or to recolor an already colored country in a different color.

Goal test A goal is reached iff all countries are colored and no two neighboring countries share the same color.

Path cost function Coloring a blank country has unit costs of one, whereas recoloring has higher costs, in order to prevent unnecessary recoloring. Recoloring is still needed, if the search depth is limited. Otherwise it would be possible to get stuck in a state that does not fulfill the goal. The cost of a path is the number of all simple coloring actions plus the number of the recoloring steps multiplied by the cost for recoloring. Thus, the cost of an optimal path is 27.

Exercise 2.2

a) Breadth-first search is a case of uniform-cost search, in which every action has the same cost. For simplicity's sake we'll assign a unit cost of one to each action. After expanding a node, the successors will have higher path costs than the neighbors of that node, because the path to those is longer. Therefore, in uniform-cost search, the successors would be added to the priority queue with a lower priority and thus be visited after all the neighbors of the initial node. This is equivalent to adding the successors to the back of a FIFO queue, which is exactly what is done in Breadth-first search.

b) Best-first search always expands the node with the best f-value, which is the value of the evaluation function $f(n)$ for a node n. For depth-first search the evaluation function $f(n)$ can be defined as the depth $d(n)$ of a given node n. Inversely, for breadth-first search it would be $f(n) = \frac{1}{d(n)}$, which leads to the node which the shallowest depth being expanded next. For uniform-cost search, $f(n) = g(n)$, with $g(n)$ being the path costs of a node n.

c) If $h(n)$ is set to a constant, than it is negligible and thus $f(n) = g(n) + h(n) = g(n)$ for A*, which is the same evaluation function as for uniform-cost search.

Exercise 2.3

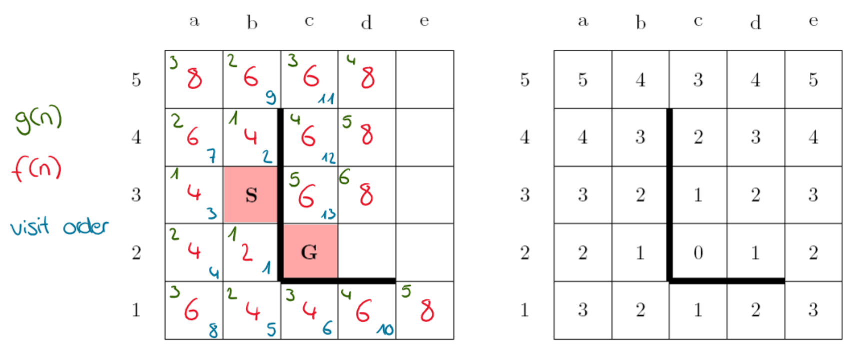
a) A $\rightarrow$ B $\rightarrow$ D

b) A $\rightarrow$ C $\rightarrow$ B $\rightarrow$ D

c) A $\rightarrow$ C $\rightarrow$ B $\rightarrow$ D

d) A heuristic h is *consistent* iff for each action x from a node a to a node b holds that $h(b) - h(a) \leq \text{cost}(x)$.

Exercise 2.4



a) The visiting order depends on the implementation and by that, is not unique.

b) The heuristic is admissible, because the Manhattan distance from a node n to the nearest goal is always less or equal than the actual path cost from the node n to the nearest goal.

c) By using the actual path cost of the optimal path as heuristic, A* would only need to expand 5 nodes. All of which with $f(n) = 6$.