Foundations of Artificial Intelligence

M. Helmert, G. Röger
J. Seipp, S. Sievers
Spring Term 2017

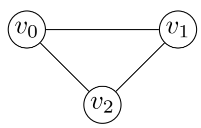
University of Basel
Computer Science

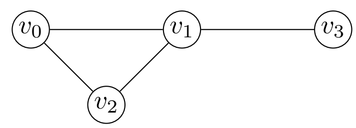
Exercise Sheet 10

Due: May 10, 2017

Exercise 10.1 (1+1.5+1.5+2 marks)

Consider the following variant of the *Seven bridges of Königsberg* problem: for a given a set of bridges that cross a river, is there a tour that starts and ends at the same location and crosses each bridge exactly once? Formally, the problem can be defined as follows: given a graph $G = (V, E)$ with set of vertices V and set of edges $E \subseteq V \times V$ and an initial vertex $v_0 \in V$, is there a sequence of vertices from V such that i) all pairs of subsequent vertices are connected with an edge from E, ii) each edge in E occurs exactly once in the sequence, and iii) the first and last vertex of the sequence is v_0. To illustrate the problem, consider the following examples:





The initial vertex is v_0 in both cases. For the graph on the left side, there is a such tour, e.g., (v_0, v_1, v_2, v_0), while there is no such tour for the graph on the right side.

(a) You can find a PDDL description of the (original instance of the) *Seven bridges of Königsberg* problem on the website of the course. The domain description (variables and actions) is given in the file `bridges-strips-domain.pddl`, and the problem description (objects, initial state and goal description) is given in the file `koenigsberg-strips.pddl`. Provide a graphical representation of the problem in the same way as the example above. Please do not forget to mark which state is the initial state.

(b) Obtain the domain-independent planning system *Fast Downward* by following the installation instructions that are given at

`http://www.fast-downward.org/ObtainingAndRunningFastDownward`.

Use *Fast Downward* with a configuration that performs greedy best-first search with the delete relaxation heuristic FF to solve the Seven bridges of Königsberg problem from the website. To do so, invoke the planner with

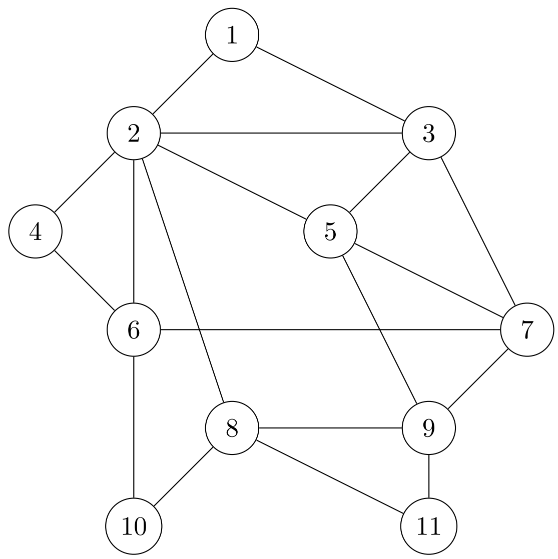
```
./fast-downward.py bridges-strips-domain.pddl koenigsberg-problem.pddl
--search "eager_greedy(ff())"
```

Is the problem solvable? If it is, provide the runtime, the number of expanded states and the plan that was found.

(c) Modify the domain description (`bridges-strips-domain.pddl`) such that it is possible to use the same bridge more than once. Solve the resulting problem with the same Fast Downward configuration that was used in Exercise 10.1 (b).

Is the problem solvable? If it is, provide the runtime, the number of expanded states and the plan that was found.

(d) Formalize the following instance of the bridges domain in PDDL and solve it with the same Fast Downward configuration that was used in Exercise 10.1 (b) where each bridge may be traversed only once. Provide the runtime, the number of expanded states and the plan that was found if the problem is solvable.



Exercise 10.2 (1+1+1 marks)
Consider the STRIPS formalization of blocks world (print-out version of slide set 34, pages 11–14). Consider the following task with blocks A, B and C, initial state $I = \{\textit{on-table}_A, \textit{on}_{B,A}, \textit{on}_{C,B}, \textit{clear}_C\}$ (left stack in the picture below) and the goal $G = \{\textit{on-table}_A, \textit{on}_{C,A}, \textit{on}_{B,C}\}$ (right stack in the picture below).

(a) Consider the STRIPS heuristic h^S (print-out version of slide set 35, page 3). Calculate the heuristic values $h^S(I)$ and $h^S(I')$ for the initial state I and the only successor state I' of I.

(b) Calculate $h^+(I)$ and $h^+(I')$.

(c) Compare and discuss the results of exercise parts (a) and (b).

Exercise 10.3 (1.5+1.5 marks)
Consider the 8-Puzzle in a STRIPS encoding $\Pi = \langle V, I, G, A \rangle$ with the following components:

- $V = \{\text{tile-at-cell}_{t,c} \mid t \in \{1, \ldots, 8\}, c \in \{(1,1), \ldots, (3,3)\}\} \cup$
  $\{\text{cell-empty}_c \mid c \in \{(1,1), \ldots, (3,3)\}\}$
- I is an arbitrary *legal* state, where a state is legal if each tile is at exactly one position, no two tiles are at the same poitions and hence there is exactly one empty position.
- $G = \{\text{tile-at-cell}_{1,(1,1)}, \ldots, \text{tile-at-cell}_{8,(3,2)}\}$
- $A = \{\textit{move}_{t,c,c'} \mid t \in \{1, \ldots, 8\}, c, c' \in \{1,2,3\} \times \{1,2,3\}, c \text{ and } c' \text{ are neibours}\}$

  All actions have cost 1 and are defined as follows:

  - $\textit{pre}(\textit{move}_{t,c,c'}) = \{\text{tile-at-cell}_{t,c}, \text{cell-empty}_{c'}\}$
  - $\textit{add}(\textit{move}_{t,c,c'}) = \{\text{tile-at-cell}_{t,c'}, \text{cell-empty}_c\}$

– $del(move_{t,c,c'}) = \{\text{tile-at-cell}_{t,c}, \text{cell-empty}_{c'}\}$

(a) For this STRIPS encoding of the 8-Puzzle, show the claim of the lecture (Chapter 35, slide 18 in the printed version): $h^+(s) \geq h^{\mathrm{MD}}(s)$ for all states s, i.e., h^+ dominates the Manhattan distance in the 8-Puzzle.

(b) Show that there exists a legal state s with $h^+(s) > h^{\mathrm{MD}}(s)$.

Important: The exercise sheets can be submitted in groups of two students. Please provide both student names on the submission.