Introduction to Artificial Intelligence

Dr. Thomas Keller | University of Zürich
C. Büchner, C. Grundke, A. Kauffmann | Spring Semester 2022

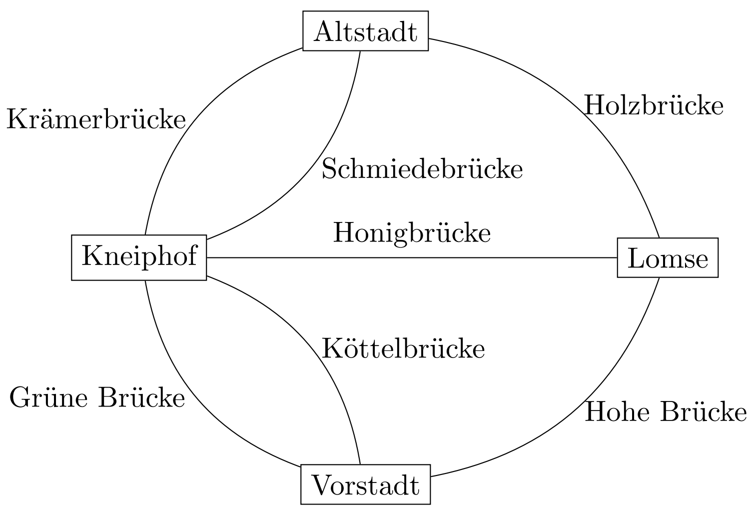
Exercise Sheet 7

Due: April 27, 2022

Points total: 20 marks

Exercise 7.1 – PDDL Formalization **5 marks**

Formalize the famous Seven Bridges of Königsberg problem in PDDL (`https://en.wikipedia.org/wiki/Seven_Bridges_of_Koenigsberg`). The problem is to find a route crossing all seven bridges *exactly* once and returning to the starting location. The graph below illustrates the land masses (nodes) and bridges (edges) of Königsberg.



(a) Provide and briefly explain the predicates you use in your formalization. (1 mark)

(b) Formalize the initial state using grounded fluents. (2 marks)

(c) Formalize the goal. Ground variables where needed. (1 mark)

(d) Formalize the (non-grounded) actions. (1 mark)

Exercise 7.2 – Abstraction Heuristics $\boxed{\textbf{6 marks}}$

Consider a planning task where an agent aims to raise a treasure. To do so, the agent must collect a key and use it to open the chest that contains the treasure. Let the problem be formalized in the SAS^+ formalism as $\Pi = \langle V, \mathrm{dom}, I, G, A \rangle$, where

- $V = \{loc, key, treasure\}$ is the set of variables with $\mathrm{dom}(loc) = \{A, B, C\}$, $\mathrm{dom}(key) = \{\top, \bot\}$, and $\mathrm{dom}(treasure) = \{\top, \bot\}$;
- $I = \{loc \mapsto B, key \mapsto \bot, treasure \mapsto \bot\}$ is the initial state;
- $G = \{key \mapsto \top, treasure \mapsto \top\}$ is the goal description; and
- $A = \{move_{A,B}, move_{B,A}, move_{B,C}, move_{C,B}, take, open\}$ is the set of actions with

$$
\begin{array}{lll}
pre(move_{A,B}) = \{loc \mapsto A\} & eff(move_{A,B}) = \{loc \mapsto B\} & cost(move_{A,B}) = 3 \\
pre(move_{B,A}) = \{loc \mapsto B\} & eff(move_{B,A}) = \{loc \mapsto A\} & cost(move_{B,A}) = 3 \\
pre(move_{B,C}) = \{loc \mapsto B\} & eff(move_{B,C}) = \{loc \mapsto C\} & cost(move_{B,C}) = 3 \\
pre(move_{C,B}) = \{loc \mapsto C\} & eff(move_{C,B}) = \{loc \mapsto B\} & cost(move_{C,B}) = 3 \\
pre(take) = \{key \mapsto \bot, loc \mapsto A\} & eff(take) = \{key \mapsto \top\} & cost(take) = 1 \\
pre(open) = \{key \mapsto \top, loc \mapsto C\} & eff(open) = \{treasure \mapsto \top\} & cost(open) = 1
\end{array}
$$

(a) Provide the state space as a graph and mark the initial state and all goal states (it consists of 12 states, some of which are not reachable from the initial state). For each state, provide the values of all variables, e.g., in the form $B\bot\bot$ for the initial state and accordingly for other states. (2 marks)

(b) Compute the projection of Π to $P = \{loc, treasure\}$ (i.e., the variable *key* is ignored). Give the abstraction that is induced by P by providing the abstract state space in the same way as in (a). (2 marks)

(c) Use the abstraction from Exercise (b) to derive a pattern database heuristic. Provide the database entries (i.e., the abstract distances for all states in the abstract state space) and use them to assign a heuristic value to each of the 12 concrete states. (2 marks)

Exercise 7.3 – Delete Relaxation Heuristics $\boxed{\textbf{9 marks}}$

Consider the STRIPS planning task $\Pi = \langle V, I, G, A \rangle$ with $V = \{a, b, c, d, e\}$, $I = \{a\}$, $G = \{d, e\}$, and $A = \{a_1, a_2, a_3, a_4\}$ with $cost = \{a_1 \mapsto 3, a_2 \mapsto 1, a_3 \mapsto 8, a_4 \mapsto 2\}$ and

$$
\begin{array}{lll}
pre(a_1) = \{a\} & add(a_1) = \{b, c\} & del(a_1) = \{\} \\
pre(a_2) = \{b, c\} & add(a_2) = \{d\} & del(a_2) = \{\} \\
pre(a_3) = \{a\} & add(a_3) = \{d, e\} & del(a_3) = \{a\} \\
pre(a_4) = \{c\} & add(a_4) = \{e\} & del(a_4) = \{c\}
\end{array}
$$

(a) Provide the relaxed planning graph for Π up to depth 2 (i.e., the resulting graph should have three variable layers and two action layers). (3 marks)

(b) Compute $h^{\max}(I)$. Provide the values for all nodes in the RPG. (2 marks)

(c) Compute $h^{\text{add}}(I)$. Provide the values for all nodes in the RPG. (2 marks)

(d) Compute $h^{\text{FF}}(I)$. Provide the marked RPG. (2 marks)

Submission rules:

- Exercise sheets must be submitted in groups of three students. Please submit a single copy of the exercises per group (only one member of the group does the submission).
- Create a single PDF file (ending `.pdf`) for all non-programming exercises. Use a file name that does not contain any spaces or special characters other than the underscore "_". If you want to submit handwritten solutions, include their scans in the single PDF. Make sure it is in a reasonable resolution so that it is readable, but ensure at the same time that the PDF size is not astronomically large. Put the names of all group members on top of the first page. Make sure your PDF has size A4 (fits the page size if printed on A4). Submit your single PDF file to the corresponding exercise assignment in MOODLE.
- For programming exercises, only create those code text files required by the exercise. Put your names in a comment on top of each file. Make sure your code compiles and test it. Code that does not compile or which we cannot successfully execute will not be graded. Create a ZIP file (ending `.zip`, `.tar.gz`, or `.tgz`; *not* `.rar` or anything else) containing the code text file(s) (ending `.py`) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly.
- Do not upload several versions to MOODLE, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.