Planning and Optimization

M. Helmert, G. Röger
R. Christen, P. Ferber, T. Keller

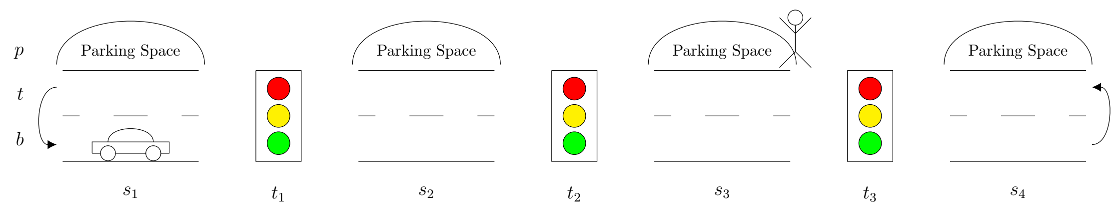
University of Basel
Fall Semester 2022

Exercise Sheet 8

Due: November 21, 2022

Important: for submission, consult the rules at the end of the exercise. Non-adherence to these rules might lead to a penalty in the form of a deduction of marks or, in the worst case, that your submission will not be corrected at all.

Exercise 8.1 (1.5+1.5+1)



In this exercise, we work with the shown driving scenario. There are four street segments s_1, s_2, s_3, and s_4 (from left to right). The agent can drive from one segment to an adjacent segment if the traffic light is green and if it is on the correct lane, (the top lane to go left, the bottom one to go right). It can switch from the top lane to the bottom lane only at segment s_1, from the bottom to the top only at segment s_4 and from the top lane to the parking space of the same segment. The agent's goal is to pick up a passenger from the parking space of segment s_3 and to be located in the parking space of segment s_4.

More formally, the scenario is defined by the following unit-cost FDR task $\Pi = \langle V, I, O, \gamma \rangle$:

$$
\begin{aligned}
V =& \{\mathit{car\text{-}seg}, \mathit{car\text{-}lane}, \mathit{pass\text{-}status}, t_1, t_2, t_3\},\\
\mathit{dom}(\mathit{car\text{-}seg}) =& \{s_1, s_2, s_3, s_4\}\\
\mathit{dom}(\mathit{car\text{-}lane}) =& \{b, t, p\}\\
\mathit{dom}(\mathit{pass\text{-}status}) =& \{\mathit{wait}, \mathit{car}\}\\
\mathit{dom}(t_i) =& \{r, y, g\} \text{ for } i \in \{1, 2, 3\}\\
I =& \{\mathit{car\text{-}seg} \mapsto s_1, \mathit{car\text{-}lane} \mapsto b, \mathit{pass\text{-}status} \mapsto \mathit{wait}, t_1 \mapsto r, t_2 \mapsto y, t_3 \mapsto r\}\\
O =& \{\langle \mathit{car\text{-}seg} = s_i \wedge \mathit{car\text{-}lane} = b \wedge t_i = g, \mathit{car\text{-}seg} := s_{i+1}, 1\rangle \mid i \in \{1, 2, 3\}\} \cup\\
& \{\langle \mathit{car\text{-}seg} = s_{i+1} \wedge \mathit{car\text{-}lane} = t \wedge t_i = g, \mathit{car\text{-}seg} := s_i, 1\rangle \mid i \in \{1, 2, 3\}\} \cup\\
& \{\langle \mathit{car\text{-}seg} = s_1 \wedge \mathit{car\text{-}lane} = t, \mathit{car\text{-}lane} := b, 1\rangle\} \cup\\
& \{\langle \mathit{car\text{-}seg} = s_4 \wedge \mathit{car\text{-}lane} = b, \mathit{car\text{-}lane} := t, 1\rangle\} \cup\\
& \{\langle \mathit{car\text{-}lane} = t, \mathit{car\text{-}lane} := p, 1\rangle\} \cup\\
& \{\langle \mathit{car\text{-}lane} = p, \mathit{car\text{-}lane} := t, 1\rangle\} \cup\\
& \{\langle \mathit{car\text{-}seg} = s_3 \wedge \mathit{car\text{-}lane} = p \wedge \mathit{pass\text{-}status} = \mathit{wait}, \mathit{pass\text{-}status} := \mathit{car}, 1\rangle\} \cup\\
& \{\langle \top, t_i := x, 1\rangle \mid i \in \{1, 2, 3\} \text{ and } x \in \{r, y, g\}\}\\
\gamma =& (\mathit{car\text{-}seg} = s_4 \wedge \mathit{car\text{-}lane} = p \wedge \mathit{pass\text{-}status} = \mathit{car})
\end{aligned}
$$

(a) Provide the syntactic projection $\Pi|_P$ of Π for the pattern $P = \{\mathit{car\text{-}seg}, \mathit{car\text{-}lane}\}$.

(b) Draw the transition system induced by the task $\Pi|_P$. Do not label the transitions. If there are multiple transitions from one state to another, draw only one arrow for all transitions. Annotate each abstract state in your transition system with its goal-distance.

(c) A PDB stores the goal distance of all abstract states in a one dimensional lookup table and uses a perfect hash function to calculate the index of a given abstract state. Provide the lookup table for pattern P from part (b) with the goal distances from part (c) and provide the function that calculates the indices that are used for the lookup.

Exercise 8.2 (2+1 marks)

(a) Prove the following claim from the lecture: Let Π be a SAS$^+$ planning task, and let P be a pattern for Π. Prove that $\mathcal{T}(\Pi|_P) \overset{G}{\sim} \mathcal{T}(\Pi)^{\pi_P}$.

Hint: It might be useful to first define the transition systems $\mathcal{T}(\Pi)$, $\mathcal{T}(\Pi|P)$, and $\mathcal{T}(\Pi)^{\pi_{\{P\}}}$. Then write down the conditions for graph-equivalence. Finally, then prove the claim by proving the conditions applied to the transition systems.

(b) Discuss the theorem you proved exercise (a). First, discuss why it is relevant. Why would we need to define $\Pi|_P$, if we already saw that π_P is a valid abstraction of $\mathcal{T}(\Pi)$, and hence we could use h^{π_P} as our heuristic? Second, discuss why is it important to exclude trivially unsolvable tasks or trivially inapplicable operators.

Exercise 8.3 (1.5+0.5+1 marks)

Consider the SAS$^+$ planning task $\Pi = \langle V, O, I, \gamma \rangle$, where

$$
\begin{aligned}
V &= \{z\} \cup \{x_i, y_i \mid 1 \le i \le 5\} \text{ with } dom(v) = \{0,1\} \text{ for all } v \in V, \\
O &= \{o_0\} \cup \{o_i, o'_i \mid 1 \le i \le 4\}, \text{where} \\
o_0 &= \langle z = 1, x_1 := 1 \land y_1 := 1 \rangle, \\
o_i &= \langle x_i = 1, x_{i+1} := 1 \rangle \text{ for } 1 \le i \le 3, \\
o_4 &= \langle x_4 = 1 \land y_4 = 1, x_5 := 1 \rangle \\
o'_i &= \langle y_i = 1, y_{i+1} := 1 \rangle \text{ for } 1 \le i \le 3, \\
o'_4 &= \langle x_4 = 1 \land y_4 = 1, y_5 := 1 \rangle \\
I &= \{z \mapsto 1\} \cup \{v \mapsto 0 \mid v \in V \setminus \{z\}\}, \text{ and} \\
\gamma &= x_5 \land y_5
\end{aligned}
$$

and the pattern collection $\mathcal{C} = \{P_1, P_2, P_3, P_4\}$ for Π, where

$$
\begin{aligned}
P_1 &= \{x_2, x_3, x_4\}, \\
P_2 &= \{x_3, y_3, x_4, x_5\}, \\
P_3 &= \{y_4, x_5\}, \text{ and} \\
P_4 &= \{x_5, y_5\}.
\end{aligned}
$$

(a) Provide the causal graph of Π. Provide a pattern collection $\mathcal{C}'$ as a simplification of $\mathcal{C}$ where patterns with causally irrelevant variables are split into patterns of causally relevant variables and where non-goal patterns are removed.

(b) Construct the compatibility graph for your collection $\mathcal{C}'$ from part (a) and determine all maximal cliques.

(c) Use your insights from part (b) to provide the canonical heuristic $h^{\mathcal{C}'}$ for your pattern collection from part (a) and simplify it with the help of the dominated sum theorem.

Submission rules:

- Exercise sheets must be submitted in groups of two or 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. Either use page numbers on all pages or put your names on each page. Make sure your PDF has size A4 (fits the page size if printed on A4).
- For programming exercises, only create those code textfiles 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.
- For the submission: if the exercise sheet does not include programming exercises, simply upload the single PDF. If the exercise sheet includes programming exercises, upload a ZIP file (ending .zip, .tar.gz or .tgz; *not* .rar or anything else) containing the single PDF and the code textfile(s) and nothing else. Do not use directories within the ZIP, i.e., zip the files directly. After creating your zip file and before submitting it, open the file and verify that it complies with these requirements.
- Do not upload several versions to ADAM, i.e., if you need to resubmit, use the same file name again so that the previous submission is overwritten.