Planning and Optimization

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

University of Basel
Fall Semester 2022

Exercise Sheet 12

Due: December 19, 2022

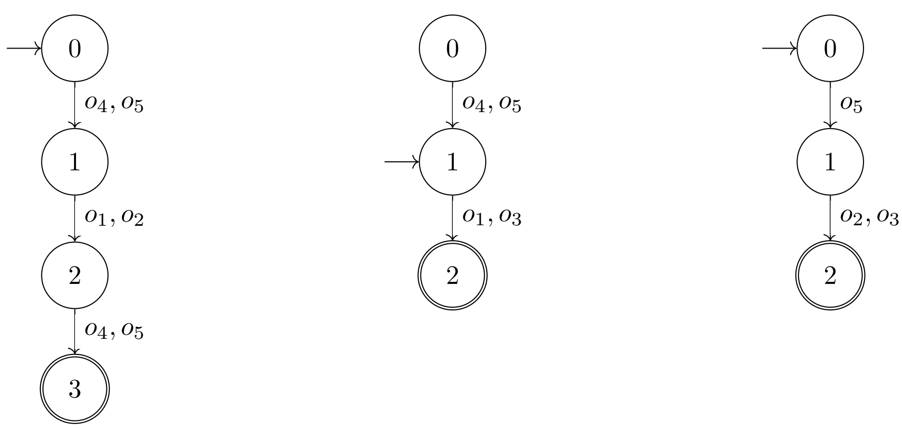
Important: for submission, consult the rules at the end of the exercise. Non-adherence to these rules might lead to a penalty in the form of a deduction of marks or, in the worst case, that your submission will not be corrected at all.

Exercise 12.1 (1+1+1+1 marks)

Consider the planning task $\Pi = \langle V, I, O, \gamma \rangle$ with

$$\begin{aligned} V &= \{a, b, c\}, \\ dom(a) &= \{0, 1, 2, 3\}, \\ dom(b) &= dom(c) = \{0, 1, 2\}, \\ I &= \{a \mapsto 0, b \mapsto 1, c \mapsto 0\}, \\ O &= \{o_1, o_2, o_3, o_4, o_5\}, \\ cost(o_i) &= 1 \text{ for } 1 \leq i \leq 4, \\ cost(o_5) &= 2, \text{ and} \\ \gamma &= (a = 3) \wedge (b = 2) \wedge (c = 2). \end{aligned}$$

The transition systems of the projections to variables are shown below: $\mathcal{T}^{\pi_{\{a\}}}$ on the left, $\mathcal{T}^{\pi_{\{b\}}}$ in the middle, $\mathcal{T}^{\pi_{\{c\}}}$ on the right.



(a) For abstraction heuristics $h^{\alpha_1}, \dots, h^{\alpha_n}$, the uniform cost partitioning is defined for all $o \in O$ and $1 \leq i \leq n$ by

$$cost_i(o) := \begin{cases} \frac{cost(o)}{n_o} & \text{if } o \text{ affects } \mathcal{T}^{\alpha_i} \\ 0 & \text{otherwise,} \end{cases}$$

where n_o is the number of abstractions α such that o affects $\mathcal{T}^{\alpha}$.

Compute the uniform cost partitioning for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ and provide the corresponding heuristic value in the initial state I.

(b) For abstraction heuristics $h^{\alpha_1}, \dots, h^{\alpha_n}$, a zero-one cost partitioning is defined for all $o \in O$ by $cost_i(o) := cost(o)$ for exactly one $i \in \{1, \dots, n\}$, and $cost_j(o) := 0$ for $j \neq i$. Compute a zero-one cost partitioning for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ which yields the maximal heuristic value in the initial state I among all zero-one cost partitionings. Provide the corresponding heuristic value in I and justify why there is no zero-one cost partitioning with higher heuristic value for I.

(c) Compute the saturated cost paritionings for the three abstraction heuristics $h^{\{a\}}, h^{\{b\}}, h^{\{c\}}$ for the orders $\langle h^{\{a\}}, h^{\{b\}}, h^{\{c\}} \rangle$ and $\langle h^{\{b\}}, h^{\{c\}}, h^{\{a\}} \rangle$. For both cost partitionings, provide both the cost and the minimal saturated cost functions for all three heuristics and provide the heuristic value of the resulting cost partitioning heuristic in I.

(d) Provide a cost partitioning that yields a higher heuristic value than the cost partitionings from (a), (b) and (c). What is the value of the corresponding heuristic in the initial state?

Exercise 12.2 (1.5+1.5 marks)
Consider a SAS$^+$ planning task $\Pi = \langle V, I, O, \gamma \rangle$ where

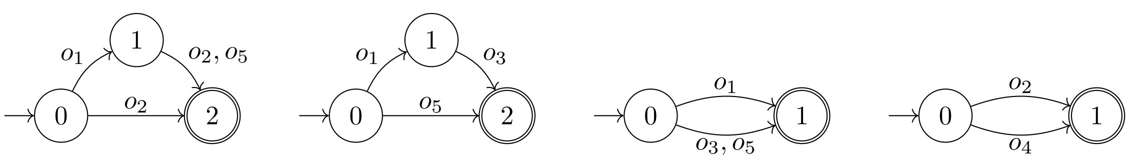
$$\begin{aligned}
V &= \{v_1, v_2, v_3\},\\
dom(v_1) = dom(v_2) = dom(v_3) &= \{A, B, C\}\\
I &= \{v_1 \mapsto C, v_2 \mapsto A, v_3 \mapsto B\},\\
O &= \{o_1, \dots, o_5\},\\
o_1 &= \langle v_1 = C \land v_2 = A, v_1 := B \land v_2 := B, 1 \rangle,\\
o_2 &= \langle v_1 = B, v_3 := A, 2 \rangle,\\
o_3 &= \langle v_1 = B \land v_3 = B, v_1 := A \land v_3 := C, 3 \rangle,\\
o_4 &= \langle v_2 = B \land v_3 = C, v_3 := A, 4 \rangle,\\
o_5 &= \langle v_3 = C, v_2 := A \land v_3 := B, 5 \rangle,\\
\gamma &= (v_1 = A) \land (v_3 = B).
\end{aligned}$$

You want to use the flow heuristic for evaluating the initial state.

(a) Convert Π to a task Π' that is in transition normal form. Use the method from the lecture that runs in linear time, but only change the domain of a variable if you need to.

(b) Specify the flow constraints for all atoms of Π'.

Exercise 12.3 (1+1+1 marks)
Consider a planning task with four variables and with operator set $O = \{o_1, o_2, o_3, o_4, o_5\}$. The cost function is defined as $cost(o_i) = i$ for $i \in \{1, \dots, 5\}$. Consider the four pattern databases induced by the four projections to the variables, which are given as follows (for brevity, self loops are omitted):



(a) What estimate do we get from the canonical heuristic for the initial state?

(b) We want to use the Post-hoc optimization heuristic. What equivalence classes of the operators do we use for the aggregation of the variables?

(c) Provide the LP solved by the Post-hoc optimization heuristic for the initial state as an input file for SoPlex. Then solve the LP and provide the objective value.

Instructions on how to install and use SoPlex are in the file `soplex-readme.txt`*.*

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.