Planning and Optimization

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

University of Basel
Fall Semester 2022

Exercise Sheet 10

Due: December 05, 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 10.1 (0.5+1+1.5 marks)

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

$$\begin{aligned} V &= \{a, b, c\}, \\ I &= \{a \mapsto \mathbf{T}, b \mapsto \mathbf{T}, c \mapsto \mathbf{F}\}, \\ O &= \{o_1, o_2\}, \\ o_1 &= \langle a, \neg b \wedge c, 2 \rangle \\ o_2 &= \langle c, b, 1 \rangle, \\ \gamma &= a \wedge b \wedge c. \end{aligned}$$

(a) Convert Π to a STRIPS planning task $\Pi' = \langle V', I', O', G' \rangle$ in set representation.

(b) Provide $R(A, O)$ for all $A \subseteq V$ with $1 \leq |A| \leq 2$.

(c) Compute $h^2(I, G)$ with the algorithm presented in Chapter F01 of the lecture. In each step, provide $h^2(I, A)$ for a subset $A \subseteq V$ of your choice that is relevant for the computation of $h^2(I, G)$.

Exercise 10.2 (1.5+1.5 marks)

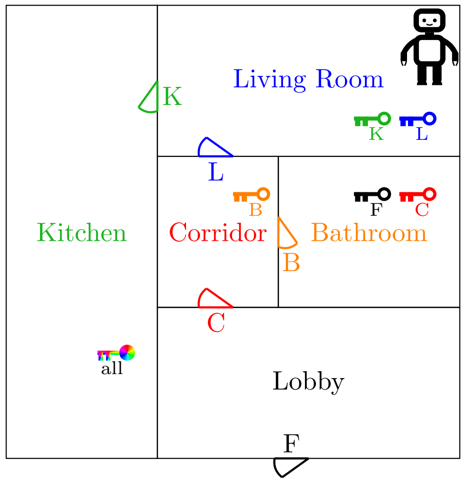
Consider the task $\Pi = \langle V, I, O, G \rangle$ with the set notation $V = \{w, x, y, z\}$, $I = \{w, x\}$, $O = \{o_1, o_2\}$, $o_1 = \langle \{x\}, \{y\}, \{w\}, 1 \rangle$, $o_2 = \langle \{y\}, \{z\}, \emptyset, 1 \rangle$, and $G = \{x, y, z\}$.

(a) Provide the task $\Pi^2 = \langle V'', I'', O'', G'' \rangle$ that results from the Π^2 compilation of Π.

(b) Use Π^2 to show that there is an admissible heuristic h such that $h^*_\Pi(I) < h_{\Pi^2}(I^2)$.

Exercise 10.3 (1 mark)

A few weeks have passed since the last incident with your household robot and you have almost forgotten its issues until you arrive at your appartement one evening just to realize that your robot has locked you out again. You swear to yourself that you'll get rid of the robot for good this time (or to at least take your own key with you in the future), but first you have to get back in. You reset your robot remotely and it starts to plan again but it doesn't start acting as you wish. Looking into the log files, you realize that parts of the memory are broken and the implemented blind search is no longer able to solve the task. You know however that you can feed constraints to your robot that must hold in every plan to decrease its memory requirements and speed up its planning process. The current situation in your home is the same as the last two times:



Provide a fact landmark of size 1 that is not true in the initial state, a disjunctive action landmark of size 2 that is tight (i.e., no subset of the disjunctive action landmark is also a disjunctive action landmark) and a network constraint. You may describe your landmarks in words or use variable and action names from the model solution of Exercise 7.1.

Exercise 10.4 (1+2 marks)

Consider the STRIPS task $\Pi = \langle V, I, O, \gamma \rangle$ with variable set $V = \{a, b, c, d, e, f, g\}$, initial state I with $I(a) = \top$ and $I(v) = \bot$ for all $v \in V \setminus \{a\}$, operator set $O = \{o_1, o_2, o_3, o_4, o_5, o_6\}$ and goal $\gamma = e$. All operators in O have cost 1 and are defined as follows:

$$
\begin{aligned}
o_1 &= \langle a, b \land \neg a \rangle \\
o_2 &= \langle a, c \rangle \\
o_3 &= \langle b, d \rangle \\
o_4 &= \langle c, d \rangle \\
o_5 &= \langle d \land g, e \land f \rangle \\
o_6 &= \langle \top, g \rangle
\end{aligned}
$$

(a) Provide the simplified relaxed task graph $sRTG(\Pi^+)$ in graphical form.

(b) Compute the set of causal fact landmarks with the fixed-point algorithm introduced in Chapter G2. You can annotate the nodes of your graph from (a) as in the lecture, but denote in which order you update the values of the nodes, and if you change the same node several times provide all intermediate values.

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.