Automated Planning (TDDD48)

Jendrik Seipp
Mika Skjelnes

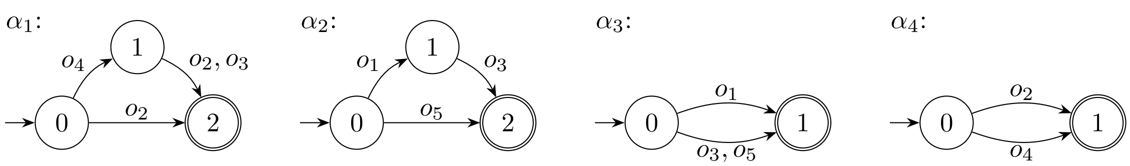
Linköping University

Lab 7

Important: For submission, consult the rules at the end of this document. Non-adherence to these rules might lead to a penalty in the form of a deduction of points. Some points are *bonus points*. These can help you reach the point quota per lab (4/12 points) and the overall point quota ($50\% \cdot 7 \cdot 12 = 42$ points).

Exercise 7.1 (1+1+1.5+1.5+1+1.5+1.5 points)

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 $i \in \{1, \ldots, 5\}$. Note that this is **not** a unit cost task. Consider the four pattern databases induced by the four projections to the variables, which are given as follows (for brevity, self loops are omitted):



Hint: We suggest to represent each cost partitioning as a table with one row for each abstraction/landmark and one column for each operator.

(a) What estimate do we get from the canonical heuristic for the initial state? (No table needed.)

(b) For abstraction heuristics $h^{\alpha_1}, \ldots, h^{\alpha_n}$, 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 $\mathcal{T}^{\alpha}$.

Provide the uniform cost partitioning for this problem. What is the heuristic value for the initial state?

(c) Provide a saturated cost partitioning for the order $\alpha_3, \alpha_1, \alpha_2, \alpha_4$. What is the heuristic value for the initial state?

(d) Provide a saturated cost partitioning for the order $\alpha_4, \alpha_3, \alpha_2, \alpha_1$. What is the heuristic value for the initial state?

(e) What estimate do we get from a heuristic that uses the maximum over all abstraction heuristics using the original cost function? How do the two orderings for saturated cost partitioning compare to this heuristic and what does this tell us about the choice of ordering for saturated cost partitioning?

(f) Consider the following set of disjunctive action landmarks based on cuts in the abstract transition systems above:

$$\mathcal{L} = \{\{o_2, o_4\}, \{o_2, o_3\}, \{o_1, o_5\}, \{o_3, o_5\}, \{o_1, o_3, o_5\}\}$$

Use SoPlex to compute an optimal cost partitioning for these landmarks. Submit both the file encoding the LP with clear names for variables and constraints, as well as the resulting cost partitioning. What is the heuristic value for the initial state?

Pull the latest revision of the labs repo to find instructions on how to use SoPlex in the file `soplex-readme.txt`.

(g) Provide the LP solved by the Post-hoc optimization heuristic for the initial state as an input file for SoPlex. Then solve the (primal) LP and provide the objective value. Finally, solve the dual LP (by using the `-y` flag) and describe the cost partitioning that the dual solution defines.

Exercise 7.2 (1.5+1.5 marks)
Consider a SAS^+ planning task $\Pi = \langle V, I, O, \gamma \rangle$ where

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

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'.

Submission rules:

- Lab sheets must be submitted in groups of 2–3 students. Clone the labs repo (`https://github.com/mrlab-ai/tddd48-labs`) and push it to a repo at the University GitLab instance `https://gitlab.liu.se`. Make sure the repo is **private** and give read access to Mika Skjelnes (`mika.skjelnes@liu.se`).
- For non-programming exercises, create a single PDF file at the location `labX/solution.pdf`. 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. 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, directly edit the code in the cloned repository and only create those code text file(s) required by the lab. 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.