Automated Planning (TDDD48)

Jendrik Seipp
Mika Skjelnes

Linköping University

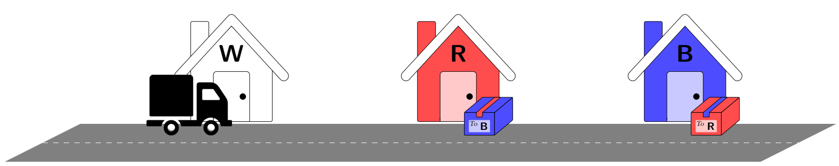
Lab 6

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 6.1 (1+1 points)

Recall the logistics problem from exercise 8.1 with the additional restriction that the truck must return to the depot W to reach the goal: Let $\Pi = \langle V, I, O, \gamma \rangle$ be a SAS$^+$ planning task with

- $V = \{t, p_B, p_R\}$ where $dom(t) = \{W, R, B\}$ and $dom(p_B) = dom(p_R) = \{W, R, B, T\}$;
- $I = \{t \mapsto W, p_B \mapsto R, p_R \mapsto B\}$;
- $O = \{move_{o,d} \mid o, d \in \{W, R, B\}, o \neq d\} \cup \{load_{p,l} \mid p \in \{p_B, p_R\}, l \in \{W, R, B\}\} \cup \{unload_{p,l} \mid p \in \{p_B, p_R\}, l \in \{W, R, B\}\}$ where
  - $move_{o,d} = \langle t = o, t := d, 1 \rangle$,
  - $load_{p,l} = \langle t = l \wedge p = l, p := T, 1 \rangle$, and
  - $unload_{p,l} = \langle t = l \wedge p = T, p := l, 1 \rangle$; and
- $\gamma = (t = W \wedge p_B = B \wedge p_R = R)$.

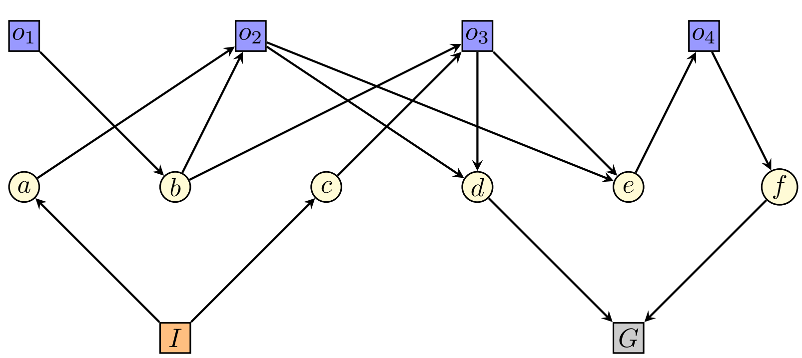


(a) Provide a fact landmark for I that neither holds in the initial state nor any goal state and justify why it holds.

(b) Provide a disjunctive action landmark for I that is a proper subset of O and justify why it holds.

Exercise 6.2 (4 points)

Consider the simplified relaxed task graph depicted below.



Compute the set of causal fact landmarks and disjunctive action landmarks of size 1 with the fixed-point algorithm introduced in the lecture. You can annotate the nodes of the above graph as in the lecture.

Bonus Exercise 6.3 (1+1+1=3 bonus points)

Let $X = \{u, v, w, x, y, z\}$ and $\mathcal{F} = \{\{u, v, x\}, \{w, x\}, \{z\}, \{u, v, y\}, \{u, v, w\}, \{u, v, x, z\}\}$.

(a) Compute a minimum hitting set H under the cost function $c = \{i \mapsto 1 \mid i \in X\}$.

(b) Compute a minimum hitting set H' under the cost function $c' = \{u \mapsto 3, v \mapsto 2, w \mapsto 1, x \mapsto 1, y \mapsto 0, z \mapsto 2\}$. What is the cost of H'?

(c) Assume an oracle can tell us all disjunctive action landmarks for all states of a given planning task. Is the minimum hitting set heuristic equivalent to the perfect heuristic in this case? Justify your answer.

Exercise 6.4 (6 points)

Consider the delete-free STRIPS planning task $\Pi^+ = \langle V, I, O, \gamma \rangle$ with

- $V = \{i, a, b, c, d, e, g\}$;
- $I = \{i \mapsto \mathbf{T}\} \cup \{v \mapsto \mathbf{F} \mid v \in V \setminus \{i\}\}$;
- $O = \{o_1, \ldots, o_7\}$ where
  - $o_1 = \langle \{i\}, \{a\}, \{\}, 1 \rangle$,
  - $o_2 = \langle \{i\}, \{b\}, \{\}, 1 \rangle$,
  - $o_3 = \langle \{a\}, \{b, c\}, \{\}, 4 \rangle$,
  - $o_4 = \langle \{b\}, \{a, c\}, \{\}, 5 \rangle$,
  - $o_5 = \langle \{a, b\}, \{d\}, \{\}, 3 \rangle$,
  - $o_6 = \langle \{c, d\}, \{e\}, \{\}, 2 \rangle$,
  - $o_7 = \langle \{d, e\}, \{g\}, \{\}, 0 \rangle$; and
- $\gamma = g$.

Compute $h^{\text{LM-cut}}(I)$ and provide all intermediate results in the same way they were given in the example of the lecture. Specifically, provide for each iteration (except the last):

- the justification graph with $h^{\max}$ annotations and marked goal zone,
- the cut,
- the cost of the cut, and
- the updated costs.

If multiple preconditions of an operator have the same maximal $h^{\max}$ value, the precondition choice function breaks ties in alphabetical order (i.e., for an operator $o = \langle \{a, b\}, \{c\}, \{\}, 2 \rangle$ with $h^{\max}(a) = 3$ and $h^{\max}(b) = 3$, we have $pcf(o) = a$). To obtain the $h^{\max}$ values, we recommend to draw a simplified relaxed task graph.

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.