

Foundations of Artificial Intelligence

M. Helmert, G. Röger
J. Seipp, S. Sievers
Spring Term 2017

University of Basel
Computer Science

Exercise Sheet 11

Due: May 17, 2017

Exercise 11.1 (0.5+1+0.5+0.5+0.5 marks)

Consider the STRIPS planning task $\Pi = \langle V, I, G, A \rangle$ with $V = \{a, b, c, d, e, f\}$, $I = \{a, b\}$, $G = \{e, f\}$, and $A = \{a_1, a_2, a_3\}$ with $\text{cost}(a) = 1$ for all $a \in A$ and

$$\begin{array}{lll} \text{pre}(a_1) = \{a\} & \text{add}(a_1) = \{c\} & \text{del}(a_1) = \{e\} \\ \text{pre}(a_2) = \{b\} & \text{add}(a_2) = \{d\} & \text{del}(a_2) = \{c\} \\ \text{pre}(a_3) = \{c, d\} & \text{add}(a_3) = \{e, f\} & \text{del}(a_3) = \{\}. \end{array}$$

Compute the following and justify your answers:

- (a) the cost of an optimal relaxed plan $h^+(I)$
- (b) the relaxed planning graph for Π up to depth 3
- (c) the maximum heuristic $h^{\max}(I)$
- (d) the additive heuristic $h^{\text{add}}(I)$
- (e) the FF heuristic $h^{\text{FF}}(I)$

Exercise 11.2 (2 marks)

Prove that h^+ is admissible. For this, prove that if Π is a STRIPS planning task and π is a plan for Π , then π^+ is a plan for the relaxed planning task Π^+ . Why does it follow from this that h^+ is admissible?

Exercise 11.3 (2+1+1 marks)

Consider a planning task where Bob aims to tighten the nuts of a gate. Initially, he is located at the shed. To be able to tighten the nuts, he needs to pick up a spanner, which is at the middle location between the shed and the gate. The problem is formalized in SAS^+ as follows. $\Pi = \langle V, \text{dom}, I, G, S \rangle$, where

- $V = \{loc, spanner, tightened\}$ is the set of variables with $\text{dom}(loc) = \{S, M, G\}$, $\text{dom}(spanner) = \{\top, \perp\}$, and $\text{dom}(tightened) = \{\top, \perp\}$;
- $I = \{loc \mapsto S, spanner \mapsto \perp, tightened \mapsto \perp\}$ is the initial state;
- $G = \{tightened \mapsto \top\}$ is the goal description; and

- $S = \{move_{S,M}, move_{M,S}, move_{M,G}, move_{G,M}, pickup, tighten\}$ is the set of actions with

$$\begin{array}{lll}
pre(move_{S,M}) = \{loc \mapsto S\} & eff(move_{S,M}) = \{loc \mapsto M\} & cost(move_{S,M}) = 3 \\
pre(move_{M,S}) = \{loc \mapsto M\} & eff(move_{M,S}) = \{loc \mapsto S\} & cost(move_{M,S}) = 3 \\
pre(move_{M,G}) = \{loc \mapsto M\} & eff(move_{M,G}) = \{loc \mapsto G\} & cost(move_{M,G}) = 3 \\
pre(move_{G,M}) = \{loc \mapsto G\} & eff(move_{G,M}) = \{loc \mapsto M\} & cost(move_{G,M}) = 3 \\
pre(pickup) = \{loc \mapsto M, & eff(pickup) = \{spanner \mapsto \top\} & cost(pickup) = 1 \\
& spanner \mapsto \perp\} \\
pre(tighten) = \{loc \mapsto G, & eff(tighten) = \{tightened \mapsto \top\} & cost(tighten) = 2 \\
& spanner \mapsto \top, \\
& tightened \mapsto \perp\}
\end{array}$$

- Provide the state space as a graph and mark the initial state and all goal states (it consists of 12 states, some of which are not reachable from the initial state). For each state, provide the values of all variables, e.g., in the form $S \perp \perp$ for the initial state and accordingly for other states.
- Compute the projection of Π to $P = \{loc, tightened\}$ (i.e., the variable *spanner* is ignored). Give the abstraction that is induced by P by providing the abstract state space in the same way as in part (a).
- Use the abstraction from part (b) to derive a pattern database heuristic. Provide the database entries (i.e., the abstract distances for all states in the abstract state space) and use them to assign a heuristic value to each of the 12 concrete states.

Exercise 11.4 (3 marks)

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

- set of variables $V = \{a, b, c, d, e, f, g\}$
- initial state $I = \{a\}$,
- goal description $G = \{g\}$, and
- set of actions $A = \{a_1, \dots, a_6\}$ with

$$\begin{array}{lll}
pre(a_1) = \{a\} & add(a_1) = \{b, d\} & cost(a_1) = 1 \\
pre(a_2) = \{b\} & add(a_2) = \{d, e, f\} & cost(a_2) = 6 \\
pre(a_3) = \{a\} & add(a_3) = \{c, d\} & cost(a_3) = 2 \\
pre(a_4) = \{c, d\} & add(a_4) = \{e\} & cost(a_4) = 1 \\
pre(a_5) = \{e\} & add(a_5) = \{f\} & cost(a_5) = 2 \\
pre(a_6) = \{d, e, f\} & add(a_6) = \{g\} & cost(a_6) = 0.
\end{array}$$

Compute $h^{LM-cut}(I)$ and provide all intermediate results in the same way they were given in the example of the lecture (including the justification graph with h^{\max} annotations).

Exercise 11.5 (1 bonus mark)

Run the following Fast Downward configurations on the “Koenigsberger Bridges Problem” from sheet 10 using the domain version in which bridges can be crossed multiple times. You will find

the domain and problem file on the course website on Thursday. Report the heuristic value for the initial state for each heuristic.

All configurations start with the prefix

```
./fast-downward.py path/to/bridges-strips-fixed-domain.pddl path/to/koenigsberg-problem.pddl
```

and differ only in the suffix `--search XXX`:

- (a) h^{\max} : `--search "astar(hmax())"`
- (b) LM-Cut heuristic: `--search "astar(lmcut())"`
- (c) h^{add} : `--search "astar(add())"`
- (d) STRIPS heuristic: `--search "astar(goalcount())"`
- (e) h^{FF} : `--search "astar(ff())"`
- (f) CEGAR heuristic: `--search "astar(cegar(subtasks=[original()]))"`
- (g) PDB heuristic with pattern containing first four variables: `--search "astar(pdb(manual([0,1,2,3])))"`
- (h) Determine h^* for the initial state.

Important: The exercise sheets can be submitted in groups of two students. Please provide both student names on the submission.