Principles of AI Planning

Prof. Dr. B. Nebel, R. Mattmüller
M. Ortlieb
Winter Semester 2013/2014

University of Freiburg
Department of Computer Science

Exercise Sheet 6

Due: Friday, December 13th, 2013

Exercise 6.1 (Finite-domain representation, 5 points)

Consider the propositional Blocksworld planning task $\Pi = \langle A, I, O, \gamma \rangle$, with

- the set of variables

$$A = \{A\text{-}clear, B\text{-}clear, C\text{-}clear, A\text{-}on\text{-}B, A\text{-}on\text{-}C, A\text{-}on\text{-}T, \\ B\text{-}on\text{-}A, B\text{-}on\text{-}C, B\text{-}on\text{-}T, C\text{-}on\text{-}A, C\text{-}on\text{-}B, C\text{-}on\text{-}T\}$$

- $I(a) = 1$ for $a \in \{B\text{-}on\text{-}T, A\text{-}on\text{-}B, A\text{-}clear, C\text{-}on\text{-}T, C\text{-}clear\}$,
  $I(a) = 0$, else.

- O contains the actions

$$\begin{aligned}
\text{move-}X\text{-}Y\text{-}Z &= \langle X\text{-}on\text{-}Y \wedge X\text{-}clear \wedge Z\text{-}clear, \\
&\qquad \neg X\text{-}on\text{-}Y \wedge Y\text{-}clear \wedge X\text{-}on\text{-}Z \wedge \neg Z\text{-}clear \rangle \\
\text{move-}X\text{-Table-}Z &= \langle X\text{-}on\text{-}T \wedge X\text{-}clear \wedge Z\text{-}clear, \\
&\qquad \neg X\text{-}on\text{-}T \wedge X\text{-}on\text{-}Z \wedge \neg Z\text{-}clear \rangle \\
\text{move-}X\text{-}Y\text{-Table} &= \langle X\text{-}on\text{-}Y \wedge X\text{-}clear, \\
&\qquad \neg X\text{-}on\text{-}Y \wedge Y\text{-}clear \wedge X\text{-}on\text{-}T \rangle
\end{aligned}$$

  for pair-wise distinct $X, Y, Z \in \{A, B, C\}$

- $\gamma = B\text{-}on\text{-}C \wedge C\text{-}on\text{-}A$.

(a) The following mutex groups can be found for Π:

$$\begin{aligned}
L_1 &= \{B\text{-}on\text{-}A, C\text{-}on\text{-}A, A\text{-}clear\} \\
L_2 &= \{A\text{-}on\text{-}B, C\text{-}on\text{-}B, B\text{-}clear\} \\
L_3 &= \{A\text{-}on\text{-}C, B\text{-}on\text{-}C, C\text{-}clear\} \\
L_4 &= \{A\text{-}on\text{-}B, A\text{-}on\text{-}C, A\text{-}on\text{-}T\} \\
L_5 &= \{B\text{-}on\text{-}A, B\text{-}on\text{-}C, B\text{-}on\text{-}T\} \\
L_6 &= \{C\text{-}on\text{-}A, C\text{-}on\text{-}B, C\text{-}on\text{-}T\}
\end{aligned}$$

Specify a planning task Π' that is equivalent to Π and in finite-domain representation by using these mutex groups. Please name the variables in a reasonable way (e.g., analogously to the examples given in the lecture).

(b) Specify the propositional planning task Π'' that is induced by Π'.

(c) A planning task $\Pi' = \langle V, I', O', \gamma' \rangle$ in finite-domain representation is equivalent to a propositional planning task Π if there is an isomorphism between $\Pi'' = \langle A'', I'', O'', \gamma'' \rangle$ and Π, where Π'' is the propositional planning task induced by Π.

There is an isomorphism between Π'' and Π if there are injective mappings $f : S \mapsto S''$ and $g : O \mapsto O''$ (where S is the set of reachable states in Π and S'' the set of states in Π'') with:

- $I'' = f(I)$
- For reachable states s_1 and s_2, if $s_2 = app_o(s_1)$ then $f(s_2) = app_{g(o)}(f(s_1))$.
- For all reachable states $s \in S$ it is true that $s \models \gamma$ iff $f(s) \models \gamma''$.

Show that the planning task Π' from exercise (b) is equivalent to Π by specifiying $f : S \mapsto S''$ and $g : O \mapsto O''$ and showing that they have the required properties.

Exercise 6.2 (PDDL, 5 points)

Every year on December 6th, Saint Nicholas, together with his companion Knecht Ruprecht[1], comes to reward good children by bringing them gifts. Unfortunately, not every child is a good child. Knecht Ruprecht is responsible for punishing naughty children. Saint Nicholas has a mysterious golden book with him, in which he can read how well every child behaved during the last year. When visiting a family, every child gets judged by Saint Nicholas based on the golden book. Every time a child was good, he gives him gifts and is very happy. Every time a child was naughty, he gets angry and Knecht Ruprecht punishes the child. If a child was neither good nor naughty, i.e., something in between, it depends on Nicholas's mood how the child gets judged. A happy Nicholas distributes gifts, whereas an angry Nicholas lets Knecht Ruprecht do his job. After rewarding a child, Nicholas is always happy and after punishing a child, he is always angry. Because Knecht Ruprecht's attendance is not needed in every case, he waits outside the house until he is called by Saint Nicholas.

Formalize this situation as a classical planning problem in `PDDL` and solve it using `pyperplan`. Assume that there are three children at pairwise different locations: *Anton*, *Bert* and *Christoph*. Bert was a *naughty* child, Christoph was *good* child and Anton was *neither a good nor a naughty* child. Formalize that punishing a child is *more expensive* than rewarding it by adding an extra operator to call Knecht Ruprecht, who waits outside.[2]. In the initial state, Nicholas and his companion are at the *North Pole* and Nicholas is *angry*. Formalize the goal such that Nicholas has to visit and judge *each child*.

Find an optimal plan and a suboptimal plan with pyperplan and explain which planning decisions makes the latter suboptimal. If you cannot find a suboptimal plan, create one manually. Send all of your files to `ortlieb@informatik.uni-freiburg.de`.

Note: Try to keep your formalization simple. Do not use negative preconditions or conditional effects, because they are not supported by `pyperplan`. Keep the domain file general enough, such that different problem files can be used, e.g., with a different number of children.

You can and should solve the exercise sheets in groups of two. You can send your solution to `ortlieb@informatik.uni-freiburg.de`. Please give both your names on your solution.

[1] `http://en.wikipedia.org/wiki/Knecht_Ruprecht`

[2] We want that the order in which Nicholas visits the children matters, which is only the case if rewarding and punishing have different costs.