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 3

Due: Friday, November 22th, 2013

Exercise 3.1 (Example for STRIPS regression, 2 points)

Consider the STRIPS planning task with atoms $A = \{a, b, c, d, e\}$, initial state $I = \{a \mapsto 0, b \mapsto 1, c \mapsto 0, d \mapsto 1, e \mapsto 1\}$, goal $\gamma = a \wedge d$, and operators $O = \{o_1, o_2, o_3\}$, where

$$o_1 = \langle b \wedge d, c \wedge e \wedge \neg d \rangle$$
$$o_2 = \langle b, a \wedge \neg c \wedge \neg d \rangle$$
$$o_3 = \langle a, d \rangle.$$

Solve this problem with a *breadth-first search* (BFS) using the STRIPS regression method. Submit the search tree that you obtain and record the solution plan. Do not expand a node further if the formula at that node is unsatisfiable or represents a set of states that is a (strict or nonstrict) subset of the set of states represented by the formula at a previously expanded node. Specify the result of regression for each node of the BFS tree.

Exercise 3.2 (General regression, 5 Points)

Old Bilbo Baggins is joyfully celebrating his eleventy-first birthday in Hobbiton. Secretly however, he is tired of the boring provincial life around him... he dreams of leaving his home and resuming his travels, just like in the old days (Oh, how he wishes to see mountains again!). Unfortunately, the festivities are packed with relatives and friends, so simply walking away is out of the question: Someone is bound to notice him sneaking out, and that would make everyone angry, which Bilbo very much wants to avoid.

$$I(p) = 1 \text{ iff } p \in \{\text{at-party}, \text{visible}\}$$

Years ago, fate brought into his possession a powerful artifact; an unimpressivly plain gold *ring* which – when worn – renders the wearer perfectly invisible! Bilbo can put it on his finger and pull it off again in every situation (it is, after all, just a ring), toggling its effect. This is modelled by the operator

$$\text{toggle-ring} = \langle \top, (\text{visible} \rhd \neg\text{visible}) \wedge (\neg\text{visible} \rhd \text{visible}) \rangle.$$

If Bilbo decides to leave the party, the outcome depends on whether he can be seen or not:

$$\text{leave-party} = \langle \text{at-party}, \neg\text{at-party} \wedge (\text{visible} \rhd \text{guests-angry}) \rangle$$

Bilbo desperately wants to leave unnoticed[1]:

$$\gamma = \neg\text{at-party} \wedge \neg\text{guests-angry}$$

With this, we have the following planning problem $\langle A, I, O, \gamma \rangle$:

$$\langle \{\text{visible}, \text{at-party}, \text{guests-angry}\}, I, \{\text{toggle-ring}, \text{leave-party}\}, \gamma \rangle$$

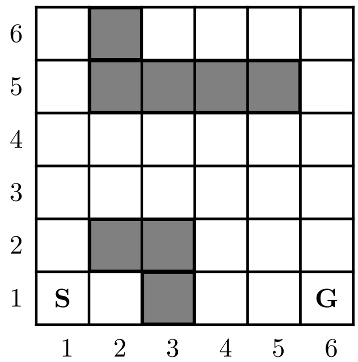
Use the regression method (full regression without splitting) to solve the problem in a *breadth-first search*. Give the resulting **search tree** with the nodes' formulae, and the **solution plan**. In every tree node, simplify the state formula as much as possible. Do not further expand a node if the

[1] For the sake of the exercise, let's just assume that Bilbo can find a quiet spot to become invisible without anyone noticing.

formula is unsatisfiable or equivalent to another (already expanded) node. In each node expansion, do **both** operator regressions, even if the first one is already fulfilled by the initial state!

Exercise 3.3 (Enforced hill-climbing, 3 points)

A robot may move on a grid in four directions, namely north, east, south and west. Consider the 6×6 board depicted below, where grey cells cannot be entered by the robot. The goal for the robot is to move from its initial position in cell $(1, 1)$ to cell $(6, 1)$.



Solve the problem with the enforced hill-climbing algorithm from the lecture using the Manhattan distance as heuristic function. The Manhattan distance d is defined as $d(x_r, y_r, x_g, y_g) = |x_r - x_g| + |y_r - y_g|$, where (x_r, y_r) and (x_g, y_g) are the coordinates of the robot and the goal respectively. There are four operators *moveNorth*, *moveEast*, *moveSouth*, *moveWest* which should be applied in exactly this order, whenever successors are generated. An operator is only applicable if the respective destination cell can be entered by the robot. There is an obvious 1-to-1 correspondence between grid cells and planning/search states, which is why we identify them in the following.

For each invocation of the `improve` procedure, mark the cell represented by the `improve`d search node with the number of the current iteration (e.g., $(1, 1)$ is marked with 1, the next cell to be improved with 2, etc.). Cross-hatch all cells that are, at some point, generated, i.e., pushed to the queue during any execution of the `improve` procedure. Finally, extract and record the solution plan.

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.