University of Lincoln
School of Computer Science
CMP2020M Artificial Intelligence
Classic AI Planning

Open the Planning Web Service at http://lcas.lincoln.ac.uk/fast-downward. For all the following tasks discuss your main findings and observations with your neighbours and the demonstrators. A planning problem is described by a “domain” specification (defining all the actions and their preconditions and effects) and a “problem” (describing objects, initial and goal states). These are given in the language PDDL, a language inspired by logic languages like LISP and Prolog.

Q1. Read the introduction on PDDL (introtopddl2.pdf) provided on blackboard.

Q2. Familiarise yourself with the interface and load the domain and problem specifications for the “blocksworld” from the pull-down menus (called “blocks”).

a. Look at the predefined *problem* and draw the *initial state* and the *goal state* defined in the problem file.
b. Find a plan yourself (not using the planner) that changes the initial state into the goal state. How long is your best plan? Is it the only possible solution to achieve the goal state? What strategy do you use to find a plan?
c. Let the web service solve the given blocks problem and compare to your own plan. Is it the same?
d. Now change the given blocks problem in the web service to represent the initial state shown in the Figure 1 and make the goal state corresponding to the Figure 2. Let the web service find a plan and check it makes sense.

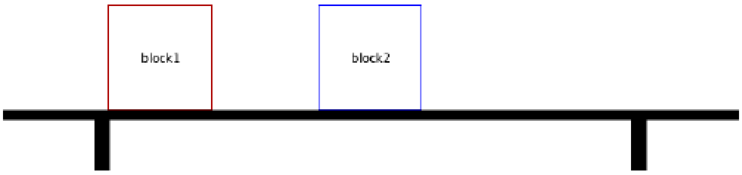


Figure 1

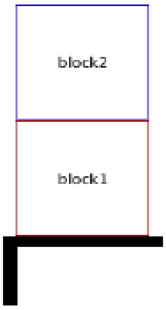
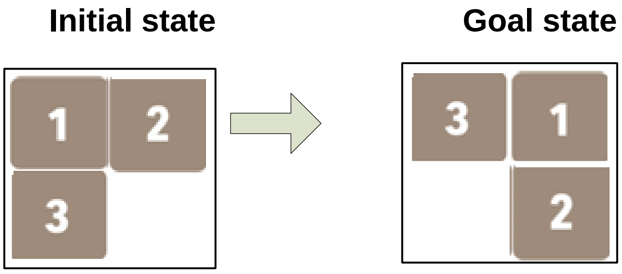


Figure 2

Q3. Look at the “robot” example on the webservice and understand it. Add another roomc, and ask the robot to put two of the balls in that room as a goal.

Q4. (*optional challenge*) Look at this classic puzzle problem:

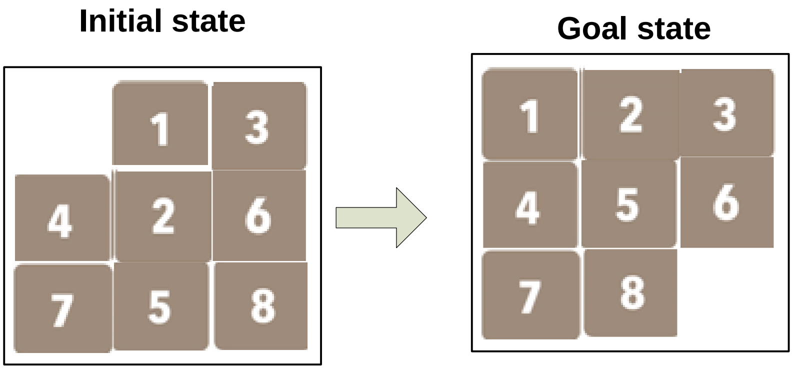


Initial state

Goal state

A problem file named “problem_puzzle” and a domain file named “domain_template_puzzle” are defined on the Web Service *(you can find them from the pull-down menus).* Read both files, understand them and define actions in the domain file which can solve the above puzzle problem as described in the problem file.

By using defined actions, define the new initial state and goal state in the “problem_puzzle” and solve a more challenging problem as follow.



Initial state

Goal state