

Programming & Written Assignment - Planning

Max points:

- CSE 4308: 100 [110 with EC]
- CSE 5360: 100 [110 with EC]

The assignment should be submitted via Canvas.

Instructions

- The answers for the TASK 1 are going to be .txt files
- The answers for the remaining TASKS can be typed as a document or handwritten and scanned.
 - Accepted document format is .pdf.
 - If you are using Word, OpenOffice or LibreOffice, make sure to save as .pdf.
 - If you are using LaTeX, compile into a .pdf file.
 - Please do not submit .txt files.
 - If you are scanning handwritten documents make sure to scan it at a minimum of 600dpi and save as a .pdf or .png file. Do not insert images in word document and submit.
- Zip all the files together as assignment4_<net-id>.zip and submit the .zip file.

Task 1 (CSE 4308: 60 Points; CSE 5360: 60 points)

The task in this programming assignment is to design appropriate descriptions of facts, actions, and goals, using the PDDL language, for two planning problems: the [Tower of Hanoi problem](#), and the 7-puzzle problem (a variation of the 8-puzzle problem where two squares are clear instead of one). You will use your descriptions as inputs to a Graphplan implementation. If your descriptions are correct, Graphplan will produce appropriate plans.

Compiling and Running the Software

The Graphplan software can be downloaded from [graphplan.zip](#). See the README file in that package for additional information. To compile the software on omega, unzip the directory, and, from that directory, type

```
make graphplan
```

Once the program compiles, it can be invoked from the commandline as follows:

```
graphplan -o [operators_file] -f [facts_file]
```

For example:

```
graphplan -o block_ops.txt -f block_facts3.txt
```

- Argument `operators_file` specifies the location of a text file containing definitions of actions. For example, see [block_ops.txt](#) for definitions of actions appropriate for the blocks world.
- Argument `facts_file` specifies the location of a text file containing definitions of facts about the environment, including objects (and types for those objects), general predicates that are always true, initial state description, and goal description. For example, see [block_facts2.txt](#), [block_facts3.txt](#), and [block_facts4.txt](#) for example fact descriptions for the blocks world.

Once you start running the software, it will ask you three questions. **Just hit enter for each question**, so as to use the default settings. If your descriptions of actions and facts are correct, the program will print out a plan achieving the stated goal.

Note that the preconds in each fact file will contain both statements that are always true in that domain (i.e., in the Tower of Hanoi domain or the 7-puzzle domain), and statements that simply describe the initial state for that specific planning problem. In addition to the facts files for the specific planning problems you are given, you will have to create a separate text file that includes all the statements that must be present in ANY facts file for that domain.

Tower of Hanoi Description

A description of the Tower of Hanoi domain can be found at [Wikipedia](https://en.wikipedia.org/wiki/Tower_of_Hanoi). In all problems that your program will be tested with there will be five discs (called disk1, disk2, disk3, disk4, disk5) and three pegs (called A, B, C). In all your facts files you will have to include both a common part (defining objects and relations among objects) and a plan-specific part (describing the initial state and goal for each plan). Note that some of the five disks may not appear in some of the planning problems.

The two planning problems you have to solve are:

Problem 1

```
initial state:
(on disk1 disk2)
(on disk2 A)
(clear disk1)
(clear B)
(clear C)
```

```
goal:
(on disk1 B)
(on disk2 C)
```

Problem 2

```
initial state:
(on disk1 disk2)
(on disk2 disk3)
(on disk3 disk4)
(on disk4 disk5)
(on disk5 C)
(clear disk1)
(clear A)
(clear B)
```

```
goal:
(on disk1 disk2)
(on disk2 disk3)
(on disk3 disk4)
(on disk4 disk5)
(on disk5 A)
```

7-puzzle Description

7-puzzle is like 8-puzzle, except that there are only pieces numbered from 1 to 7 (not from 1 to 8), and there are two clear squares on the board. At any move, we can move a numbered piece to an adjacent clear square.

The two planning problems you have to solve are (X indicates a clear square):

Problem 1

initial state:

123
456
7XX

goal:

123
X56
4X7

Problem 2

initial state:

XX7
654
321

goal:

123
456
7XX

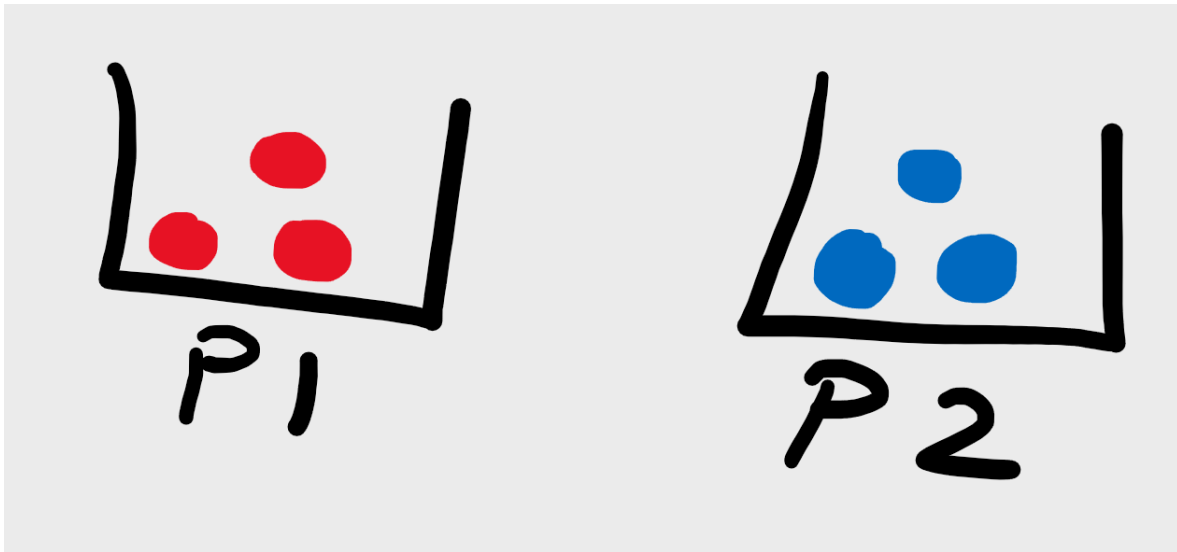
Grading

This task will be graded for 60 points. Half the points will correspond to your solutions for the Tower of Hanoi world, and the rest will correspond to your solutions for the 7-puzzle problem. Specifically, the point allocation is:

- 30 points: defining facts and actions correctly. The language that you define (i.e., the actions, objects, and general statements that are always true) should be sufficient not only for the specific plans that you are required to construct, but also for any other planning problems that we can define in the Tower of Hanoi domain or the 7-puzzle domain. As part of grading, we will also test your solutions on planning problems that we will make up.
- 30 points: solving the planning problems you are given (two for the Tower of Hanoi domain, two for the 7-puzzle domain) + 1 additional problem per domain. You get 5 points for each problem. If you solve all 6 correctly, you get 30 points.

Task 2 (CSE 4308: 15 Points (+10 Points EC); CSE 5360: 15 points (+10 Points EC))

Consider the following scenario:



Your task is to get all the blue marbles in P1 and red marbles in P2

The actions available are as follows:

- You can move 2 marbles from P1 to P2 if they are each of different colors (one blue or one red)
- You can move 2 marbles from P2 to P1 if they are of the same color (either you can move 2 red or you can move 2 blue)

Give the PDDL description to represent the above as a Planning Problem. Do not forget to also define and describe the predicates and constants that you are going to use.

Extra Credit (10 pts): Also, give a complete plan (using the actions described) for getting from the start to the goal state

Task 3 (CSE 4308: 10 Points (EC); CSE 5360: 10 points)

Consider the problem in Task 2. Let us say that, when you move the marbles, it is possible to sometimes end up moving only one of them (You try to move 2 Red marbles from P2 to P1 but only end up moving one). How would you modify the actions you described in Task 2 to account for this if you were going to try and handle this scenario by

- Execution Monitoring/Online Replanning
- Conditional Planning

In both cases, show what the modifications are (If no modification is necessary, explain why).

Task 4 (CSE 4308: 15 Points; CSE 5360: 15 points)

Suppose that we are using PDDL to describe facts and actions in a certain world called JUNGLE. In the JUNGLE world there are 4 predicates, each predicate takes at most 3 arguments, and there are 5 constants. Give a reasonably tight bound on the number of unique states in the JUNGLE world. Justify your answer.

