Artificial Intelligence: Assignment 4

Due date: April 2 (Wednesday)

Problem 1 (4 points)
Read Chapters 11 and 12, and the PRODIGY article, and answer the following questions:

(a) What is the difference between operators and inference rules?

(b) Can a partial-order planner handle inference rules? If it can, briefly explain how you can modify the POP algorithm to use inference rules. If not, discuss the main obstacles.

(c) The textbook describes hierarchical decomposition in partial-order planning (Sections 12.2 and 12.3). Suppose that you need to add a similar technique to PRODIGY, and integrate it with control rules. Identify the main difficulties and briefly discuss ways to overcome them.

Problem 2 (6 points)
Implement a program for constructing plans in the Robot Domain with two operators:

$go(x, y)$ x, y: room *Pre:* robot-in(x) $\wedge$ (door(x, y) $\vee$ door(y, x)) *Eff:* del robot-in(x); add robot-in(y)	$push(b, x, y)$ b: box; x, y: room *Pre:* in(b, x) $\wedge$ robot-in(x) $\wedge$ (door(x, y) $\vee$ door(y, x)) *Eff:* del in(b, x); del robot-in(x); add in(b, y); add robot-in(y)

Your program should read an initial state and goals from a file, and print out a plan that achieves these goals; all test problems will be solvable. Your program may return a non-optimal plan; however, if it finds a shortest plan, you will get a bonus. The format for encoding the initial state and goals is as follows:

```
rooms: <room-name> ... <room-name>
boxes: <box-name> ... <box-name>

<init-literal>
   ...
<init-literal>

<goal-literal>
   ...
<goal-literal>
```

Every room name and every box name is a string of two lower-case letters; all names are distinct. The literals after the first blank line describe the initial state, and the literals after the second blank line are the goals. An initial-state literal may be "`door <room-name> <room-name>`", "`in <box-name> <room-name>`", or "`robot-in <room-name>`". A goal literal is either "`in ...`" or "`robot-in ...`". For example, the following literals encode the initial state in the picture and the goal of moving both boxes into room `rc`:

```
rooms: ra rb rc
boxes: ba bb

door ra rb
door rb rc
in ba ra
in bb rb
robot-in rc

in ba rc
in bb rc
```