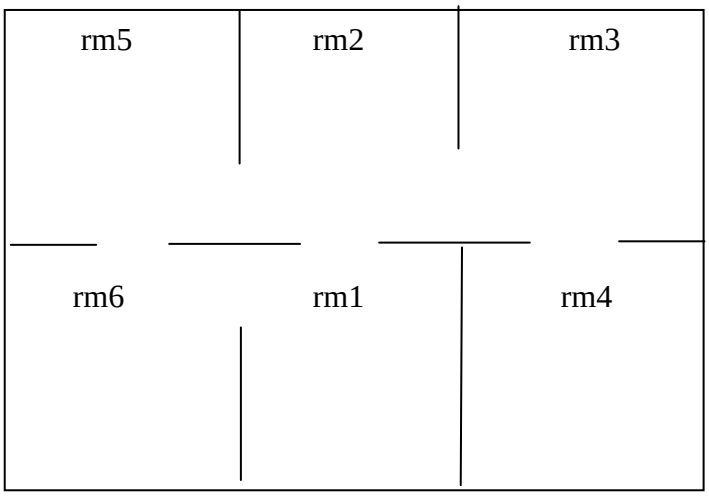
6.034 Recitation 1: Rules Problem 2 (9/17/04)

LOrtiz (Orig. KKoile)

1. Forward Chaining

Consider a robot that moves around in the following environment and figure out where he goes.



The connectivity between the rooms is described by a set of assertions for a rule-based system indicating what rooms are connected by doors (shown above as gaps):

(door rm1 rm2)
(door rm2 rm5)
(door rm2 rm3)
(door rm4 rm3)
(door rm5 rm6)
(door rm6 rm1)

In addition there is an assertion indicating the current position of the robot:
(loc rm1)

You are given the following rules:

```
(GO     if (loc ?x)
           (door ?x ?y)
        add (loc ?y)
        delete (loc ?x))

(STOP   if (loc rm4)
        add (stop))
```

A. Fill in the following sequence of (loc …) assertions that would result from running these rules with the assertions given above.

Step	Room robot is in
1.	rm1
2.	
3.	
4.	
5.	
6.	

B. How would the behavior of the system change if we add the following rule after the STOP rule:

```
(SYM    if (door ?r ?s)
        add (door ?s ?r))
```

C. Does moving the SYM rule before the GO rule change the behavior of the system?

D. How does the sequence of rooms differ, both without SYM and with SYM at end, if the assertion list is reordered:

```
(door rm1 rm2)
(door rm4 rm3)
(door rm2 rm3)
(door rm2 rm5)
(door rm5 rm6)
(door rm6 rm1))
```

Step	Room robot is in (without SYM)	Room robot is in (with SYM)
1.	rm1	rm1
2.		
3.		
4.		
5.		
6.		

2. Backward Chaining

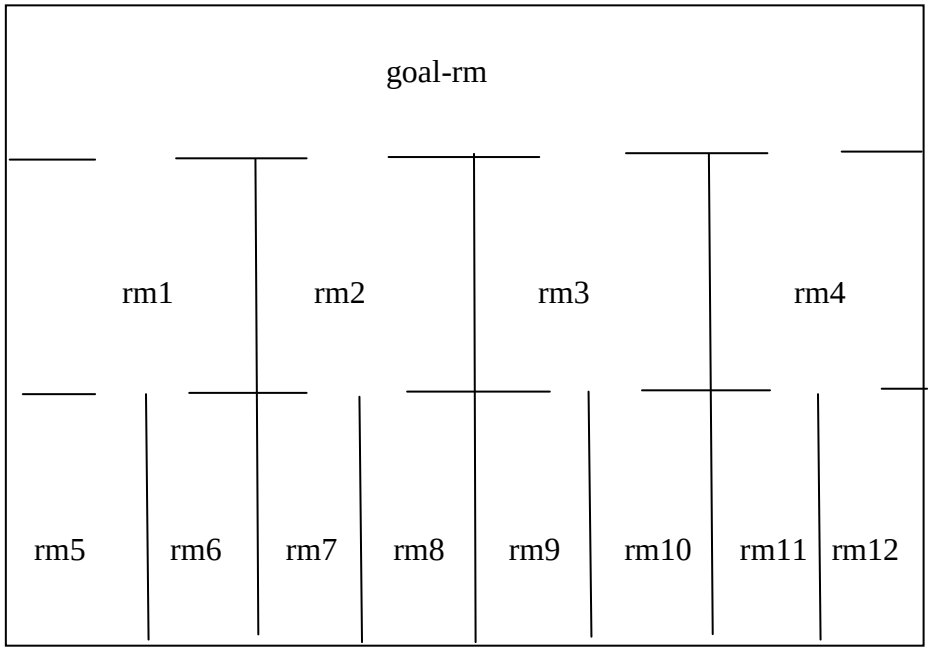
Now let's try inferring how to get a robot from its current location to the goal room.

The rules are:

```
(Rule1    if (door ?x ?y)
             (currently-in ?x)
          then (get-to ?y))

(Rule2   if (door ?x ?y)
            (get-to ?x)
          then (get-to ?y)))
```

The room layout is shown below, along with the set of assertions used to represent it, and the assertion indicating the robot's location. Assume the rules and assertions are used in the order shown. If an assertion is not in the database, you can answer "no" when the system asks about that assertion.




```
(door rm1 goal-rm)
(door rm2 goal-rm)
(door rm3 goal-rm)
(door rm4 goal-rm)
(door rm5 rm1)
(door rm6 rm1)
(door rm7 rm2)
(door rm8 rm2)
(door rm9 rm3)
(door rm10 rm3)
(door rm11 rm4)
(door rm12 rm4)
(currently-in rm12)
```

A. The system now tries to infer by backward chaining (get-to goal-rm). Draw a tree showing the sequence of subgoals produced, show the database queries, and show the questions that the system would generate. (You may stop after showing what happens as a consequence of the first 8 questions.)

B. This problem doesn't seem particularly well-suited to backward chaining. Why not?