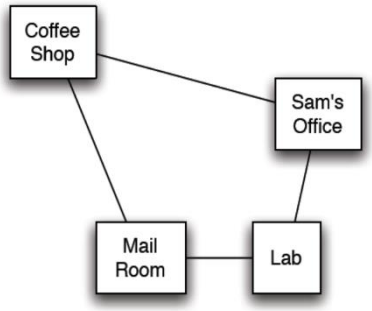
Question from CMPT 310 D100 at Simon Fraser University
Author of Question: Dr. Hazra Imran

The problem is to solve the delivery robot example using planning as CSP. Remember that Rob is a delivery robot who navigates in the following environment:



The state is defined by the following variables/features:

- **RLoc** - Rob's location
  - Domain: Coffee Shop (*cs*), Sam's office (*off*), Mail Room (*mr*), or Laboratory (*lab*)
- **RHC** - Rob has coffee.
  - Domain: True or False
- **SWC** - Sam wants coffee.
  - Domain: True or False
- **MW** - Mail is waiting.
  - Domain: True or False
- **RHM** - Rob has mail.
  - Domain: True or False

The actions are defined by the following preconditions and results:

Action	Preconditions	Effects/ Results
Move Rob's move action: Move clockwise (***mc***), Move counter-clockwise (***mcc***)		
PUC: Rob picks up coffee	Rob must be at the coffee shop	**Rob has coffee**
DelC: Rob delivers coffee	Rob must be at the office and must have coffee	**Rob does not have coffee, Sam does not want coffee**
PUM: Rob picks up mail	Rob must be in the mail room, and mail must be waiting	**Rob has mail, mail is not waiting**
DelM: Rob delivers mail	Rob must be at the office and have mail	**Rob does not have mail**

You need to reformulate the STRIPS model as CSPs with different horizons, and then solve them by implementing the Arc Consistency + Domain Splitting algorithm. The pseudocode is provided in the below:

```
solved = false
horizon = 0
while solved = false
     map STRIPS into CSP with horizon
     solve CSP → solution
     if solution then
        solved = T
     else
        horizon = horizon + 1
Return solution
```

Remember that if the horizon is *k*, you need to define a CSP variable for each STRIPS variable at time step 0 to *k*, and for each STRIPS action at time step 0 to *k* - 1. You also need CSP constraints for start and goal values, as well as preconditions and effects of actions. Another constraint that you have to consider is that at each time step only one action can be done (mutual exclusion.)

The initial state is **RLoc**: *off*, **RHC**: *F*, **SWC**: *T*, **MW**: *F*, and **RHM**: *T*.
The goal state is **RLoc**: *off* and **SWC**: *F*.

The program should output the horizon of the CSP that it has solved. It should also print all the variable assignments in the solution.

For example:

Horizon: 4
RLoc0 = off
RHC0 = F
etc…
Move0 = T
PUC0 = F
etc…
RLoc1 = cs
RHC1 = F
etc…