LAB 6

Read C9 p. 22-28 in advance!

The following rules are given:

Rules:

1.If the service is poor or the food is rancid then the tip is cheap.
2.If the service is good then the tip is normal.
3.If the service is excellent or the food is delicious then the tip is generous.

You must define the degree curves for the predicates: *poor*, *good*, *excellent*, *rancid*, *delicious*, *cheap*, *normal* and *generous*.

1. The program interface should ask the user what are his/her ratings for service and food (on a 10-point scale) and it should return a recommendation for the tip (as a percentage between 0-25).

2. We assume that the premises of a rule are connected with only one type of connector (or/and).
   The rules will be represented in any format you consider appropriate.
   Just as a suggestion, the rule "If the service is poor or the food is rancid then the tip is cheap" may be written as
   [or , [service/poor, food/rancid] , [tip/cheap]]

3. Only the user's ratings will be given at the console/GUI; the rules must be read from a file.

4. The reasoning procedure will be implemented in the version presented at the course C9 p.25 (from Ronald Brachman, Hector Levesque. Knowledge representation and reasoning, Morgan Kaufmann 2004).
   For the last step of "Defuzzification", the aggregated degree curve may be discrete (computed only for the inputs 0, 1,…, 25).

5. The program should run (that means asking for user's ratings and providing the output) repeatedly until "stop" is written in the console/GUI.

Suggestion in PROLOG to define and deal with the degree curves:

```
%defined in the .pl file
poor(X,0):-X>=0,X<0.2.
poor(X,Y):-X>0.2,Y is X+1.
%...


?- poor(3,P).
P = 4.

?- call(poor,3,P).
P = 4.

?- L=[service/poor],member(X/Y,L),call(Y,3,P).
L = [service/poor],
X = service,
Y = poor,
P = 4.
```