Lab Report-Artificial Intelligence Lab

ID:171311066

Software Environment: Swi Prolog

Table of Contents

[Introduction: 1](#_Toc34850912)

[Facts 1](#_Toc34850913)

[Rules 1](#_Toc34850914)

[problem 1:family relation facts. 2](#_Toc34850915)

## Introduction:

## Facts

A **fact** is a predicate expression that makes a declarative statement about the problem domain. Whenever a variable occurs in a Prolog expression, it is assumed to be **universally quantified**. Note that all Prolog sentences must end with a period.

Ex: likes(mohan, sabila). /\* mohan likes sabila \*/

## Rules

A **rule** is a predicate expression that uses logical implication (:-) to describe a relationship among facts. Thus a Prolog rule takes the form

Examples of valid rules:

friends(X,Y) :- likes(X,Y),likes(Y,X) /\* X and Y are friends if they like each other \*/

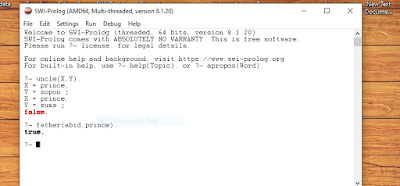
Examples of invalid rules:

left\_of(X,Y) :- right\_of(Y,X) /\* Missing a period \*/

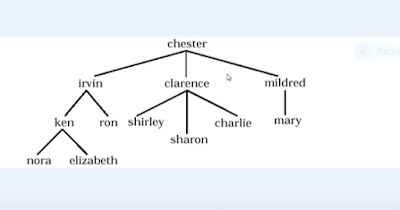
## problem 1:family relation facts.

male(abid).  
male(prince).  
male(sakib).  
male(sopon).  
female(suma).  
female(riya).  
female(nipa).  
parents(abid,prince).  
parents(abid,riya).  
parents(prince,ripa).  
parents(prince,sakib).  
parents(riya,sopon).  
parents(riya,suma).  
  
father(X,Y):-parents(X,Y),male(X).  
mother(X,Y):-parents(X,Y),female(X).  
sibling(X,Y):-parents(Z,X),parents(Z,Y),X\=Y.  
brother(X,Y):-parents(Z,X),parents(Z,Y),sibling(X,Y),female(X).  
grandfather(X,Y):-parents(X,Z),parents(Z,Y),male(X).  
grandmother(X,Y):-parents(X,Z),parents(Z,Y),female(X).  
uncle(X,Y):-parents(Z,Y),sibling(X,Z),male(X).  
aunt(X,Y):-parents(Z,Y),sibling(X,Z),female(X).  
cousin(X,Y):-grandfather(Z,X),grandfather(Z,Y),X\=Y.

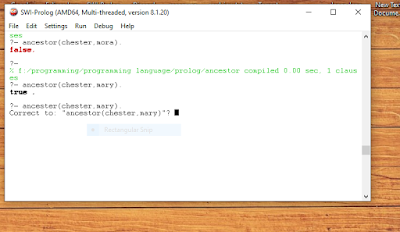
Output:



Problem 2: how to find ancestor



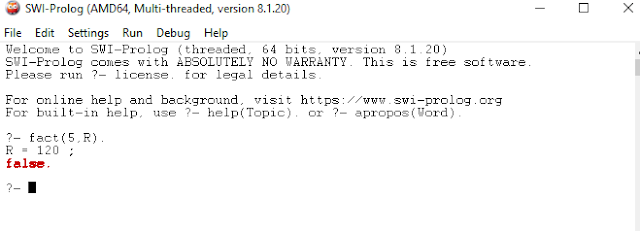
#facts are  
  
  
parent(chester,irvin).  
parent(cester,clarence).  
parent(chester,mildred).  
parent(irivin,ken).  
parent(irivin,ron).  
parent(clarence,shirley).  
parent(clarence,sharon).  
parent(clarence,charlie).  
parent(mildred,mary).  
parent(ken,mora).  
parent(ken,elizabeth).  
  
#rules:  
ancestor(X,Y):-  
    parent(X,Y).  
  
ancestor(X,Y):-  
    parent(X,Z),  
    ancestor(Z,Y).  
  
  
  
  
Output:



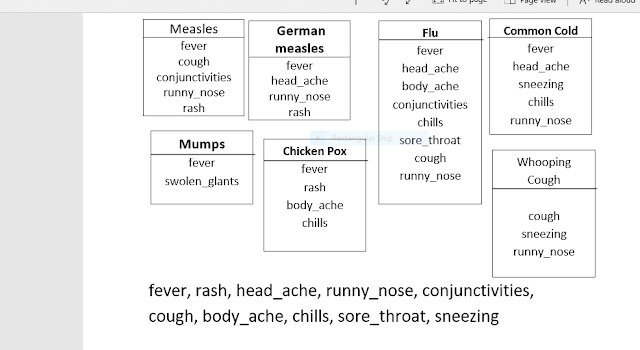
Problem 03 : find factorial problem.

fact(0,1).  
fact(N,F):-  
(  
  
 % The below is for +ve factorial.  
 N>0 ->  
 (  
  N1 is N-1,  
  fact(N1,F1),  
  F is N\*F1  
 )  
 ;  
  
 % The below is for -ve factorial.  
 N<0 ->  
 (  
  N1 is N+1,  
  fact(N1,F1),  
  F is N\*F1  
 )  
).

Output:



Problem 4:Disease problem.



Rules:-  
  
hypothesis(X,chicken pox):- symptom(X, fever), symptom(X, rash),symptom(X,body\_ache), symptom(X, chills).  
  
  
domains  
    disease,indication,name = symbol  
  
predicates  
    hypothesis(name,disease)  
    symptom(name,indication)  
  
clauses  
    symptom(amit,fever).  
    symptom(amit,rash).  
    symptom(amit,headache).  
    symptom(amit,runn\_nose).  
  
    symptom(kaushal,chills).  
    symptom(kaushal,fever).  
    symptom(kaushal,hedache).  
  
    symptom(dipen,runny\_nose).  
    symptom(dipen,rash).  
    symptom(dipen,flu).  
  
  
    hypothesis(Patient,measels):-  
        symptom(Patient,fever),  
        symptom(Patient,cough),  
        symptom(Patient,conjunctivitis),  
        symptom(Patient,rash).  
  
    hypothesis(Patient,german\_measles) :-  
        symptom(Patient,fever),  
        symptom(Patient,headache),  
        symptom(Patient,runny\_nose),  
        symptom(Patient,rash).  
  
    hypothesis(Patient,flu) :-  
        symptom(Patient,fever),  
        symptom(Patient,headache),  
        symptom(Patient,body\_ache),  
        symptom(Patient,chills).  
  
    hypothesis(Patient,common\_cold) :-  
        symptom(Patient,headache),  
        symptom(Patient,sneezing),  
        symptom(Patient,sore\_throat),  
        symptom(Patient,chills),  
        symptom(Patient,runny\_nose).  
  
    hypothesis(Patient,mumps) :-  
        symptom(Patient,fever),  
        symptom(Patient,swollen\_glands).  
  
    hypothesis(Patient,chicken\_pox) :-  
        symptom(Patient,fever),  
        symptom(Patient,rash),  
        symptom(Patient,body\_ache),  
        symptom(Patient,chills).

output:-

?-  symptom(Patient,fever).

Patient = kaushal.