ARTIFICIAL INTELLIGENCE

LAB 2

AIM: Study of RULES & UNIFICATION

EXERSICE:

1. Write a prolog program for the following facts and rules and answer the given question

i. Parva has symptom fever
ii. Parva has symptom rash
iii. Parva has symptom headache
iv. Parva has symptom runny nose
v. Vidhi has symptom chills
vi. Vidhi has symptom fever
vii. Vidhi has symptom headache
viii. Vivan has symptom runny nose
ix. Vivan has symptom rash
x. Vivan has symptom flu

Rule 1: Patient has Disease measles if Patient has symptoms fever, cough, conjunctivitis and rash.
Rule 2: Patient has Disease german measles if Patient has symptoms fever, headache, runny nose and rash.
Rule 3: Patient has Disease flu if Patient has symptoms fever, headache, body-ache and chills.
Rule 4: Patient has Disease common cold if Patient has symptoms headache, sneezing, sore throat, chills and runny nose.
Rule 5: Patient has Disease mumps if Patient has symptoms fever and swollen glands.
Rules 6: Patient has Disease chicken pox if Patient has symptoms fever, rash, body-ache and chills.

Question: Identify patient with any particular disease based on rules and facts given above.

Solution:

Code:

```
domains
    patient,indication,disease=symbol
predicates
    symptom(patient,indication).
    hypothesis(patient,disease).
clauses
    symptom(parva,fever).
    symptom(parva,rash).
    symptom(parva,headache).
    symptom(parva,runny_nose).
    symptom(vidhi,chills).
    symptom(vidhi,fever).
    symptom(vidhi,headache).
    symptom(vivan,runny_nose).
    symptom(vivan,rash).
    symptom(vivan,flu).
```

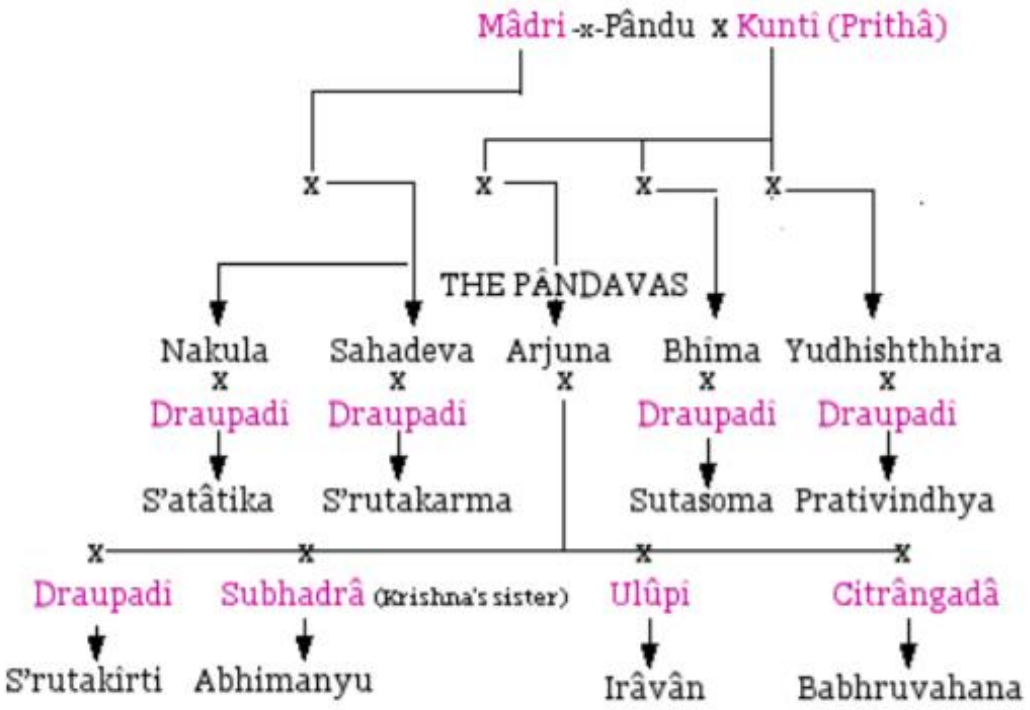
```
hypothesis(Patient,measles):-
    symptom(Patient,fever),
    symptom(Patient,cough),
    symptom(Patient,conjuntivitis),
    symptom(Patient,rash).

hypothesis(Patient,german_measles):-
    symptom(Patient,fever),
    symptom(Patient,headache),
    symptom(Patient,ranny_nose),
    symptom(Patient,rash).

hypothesis(Patient,flu):-
    symptom(Patient,fever),
    symptom(Patient,headache),
    symptom(Patient,body_ache),
    symptom(Patient,chills).

hypothesis(Patient,comman_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,chiken_pox):-
    symptom(Patient,fever),
    symptom(Patient,rash),
    symptom(Patient,body_ache),
    symptom(Patient,chills).
```

Output:

```
Goal: hypothesis(X,measles)
No Solution

Goal: hypothesis(X,german_measles)
X=parva
1 Solution

Goal: hypothesis(X,flu)
No Solution

Goal: hypothesis(X,cold)
No Solution

Goal: hypothesis(X,mumps)
No Solution

Goal: hypothesis(X,chicken_pox)
No Solution
```

2. Write a program for family tree given below which contains three predicates: male, female, parent. Make rules for family relations: father, mother, grandfather, grandmother, brother, sister, uncle, aunt, nephew and niece.



Solution:

Code:

```
predicates
    male(symbol).
    female(symbol).
    parent(symbol,symbol).
    father(symbol,symbol).
    mother(symbol,symbol).
    wife(symbol,symbol).
    grandfather(symbol,symbol).
    grandmother(symbol,symbol).
    brother(symbol,symbol).
    sister(symbol,symbol).
    uncle(symbol,symbol).
    aunt(symbol,symbol).
    nephew(symbol,symbol).
    niece(symbol,symbol).
```

```
clauses
    male("Pandu").
    male("Nakula").
    male("Sahadeva").
    male("Arjuna").
    male("Bhima").
    male("Yudhishthira").
    male("Satanika").
    male("Shrutasena").
    male("Shrutakarma").
    male("Abhimanyu").
    male("Iravan").
    male("Babruvahana").
    male("Sutasoma").
    male("Prativindhya").

    female("Madri").
    female("Kunti").
    female("Draupadi").
    female("Subhadra").
    female("Ulupi").
    female("Chitrangada").
```

```
    parent("Pandu","Nakula").
    parent("Pandu","Sahadeva").
    parent("Pandu","Arjuna").
    parent("Pandu","Bhima").
    parent("Pandu","Yudhishthhira").
    parent("Madri","Nakula").
    parent("Madri","Sahadeva").
    parent("Kunti","Arjuna").
    parent("Kunti","Bhima").
    parent("Kunti","Yudhishthira").
    parent("Nakula","Satanika").
    parent("Draupadi","Satanika").
    parent("Sahadeva","Shrutasena").
    parent("Draupadi","Shrutasena").
    parent("Arjuna","Shrutakarma").
    parent("Arjuna","Abhimanyu").
    parent("Arjuna","Iravan").
    parent("Arjuna","Babruvahana").
    parent("Draupadi","Shrutakarma").
    parent("Subhadra","Abhimanyu").
    parent("Ulupi","Iravan").
    parent("Chitrangada","Babruvahana").
    parent("Bhima","Sutasoma").
    parent("Draupadi","Sutasoma").
    parent("Yudhishthira","Prativindhya").
    parent("Draupadi","Prativindhya").
```

```
father(X,Y):-
    parent(X,Y),
    male(X).
mother(X,Y):-
    parent(X,Y),
    female(X).
wife(X,Y):-
    parent(X,Z),
    parent(Y,Z),
    male(X),
    female(Y).
grandfather(X,Y):-
    father(X,Z),
    father(Z,Y).
grandmother(X,Y):-
    mother(X,Z),
    father(Z,Y).
brother(X,Y):-
    father(A,X),
    father(A,Y),
    mother(B,X),
    mother(B,Y),
    male(X),
    not(X=Y).
sister(X,Y):-
    father(A,X),
    father(A,Y),
    mother(B,X),
    mother(B,Y),
    female(X),
    not(X=Y).
```

```
uncle(X,Y):-
    father(Z,Y),
    brother(X,Z).
aunt(X,Y):-
    father(Z,Y),
    brother(B,Z),
    wife(B,X).
nephew(X,Y):-
    father(Z,Y),
    brother(X,Z),
    male(X),
    male(Y).
niece(X,Y):-
    father(Z,Y),
    brother(X,Z),
    male(X),
    female(Y).
```

Output:

```
Goal: father(X,arjuna)
X= pandu 1 Solution

Goal: grandfather(X,abhimanyu)
X=pandu
1 Solution

Goal: uncle(X,babhruvahana)
X=nakula
X=sahadev
X=bhima
X=yudhishthhira
4 Solutions

Goal: parent(X,satatika)
X=nakula
X=draupadi

Goal: aunt(X,iravan)
No Solution

Goal: mother(X,abhimanyu)
X=subhadra
1 Solution
```

3. Write a prolog program for the following facts and rules, and trace the given goals:

i. hardware is easy course
ii. Books for hardware are available
iii. logic is not easy course
iv. graphics is easy course
v. graphics has 8 credits
vi. graphics has lab component
vii. Books for database are available
viii. Mary takes compilers

Rule1: X takes Y, if Y is easy course and books for Y are available.
Rule2: X takes Y, if Y has 8 credits and Y has lab component.

Solution:

Code:

```
domains
    name,level=symbol
predicates
    course(name,level).
    book(symbol,symbol).
    has(name,symbol).
    takes(symbol,symbol).
    take_course(symbol,symbol).
clauses
    course(hardware,easy).
    course(logic,not_easy).
    course(graphic,easy).
    book(hardware,available).
    book("database",available).
    has(graphic,"8").
    has(graphic,lab_component).
    takes(mary,compiler).

take_course(Cname,Course):-
    course(Course,easy),
    book(Course,available).

take_course(Cname,Course):-
    has(Course,"8"),
    has(Course,lab_component).
```

Goals:

a) Does Mary take a graphics course?
take_course(mary,graphics)
No

b) Which course Mary takes?
take_course(mary,X)
X=hardware
X=graphic
2 Solutions

c) Who takes graphics courses?
take_course(X,graphic)
No Solution