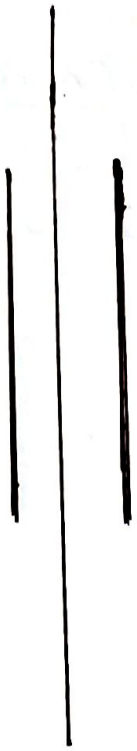
KATHMAND UNIVERSITY



LAB-2

From
Name: Aron Shrestha
Class: 3rd year 1st Sem
Roll no: 42

To
Name: Nabin Ghimire
Department of CSE

Date: 2021/06/19

LAB 2

Facts of previous lab

```
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pal).
parent(pat,jim).

female(pam).
female(liz).
female(ann).
female(pat).
male(bob).
male(tom).
male(jim).
```

In this lab we explore rule.

Rule:

A predicate expression that uses logical implication (:-) to describe a relationship among facts is known as a rule.

Rules describe in our lab

① offspring(Y,X) :- parent(X,Y).

/* since if X is parent of Y then our rule understands Y is offspring of X */

Example query

?- offspring(liz, tom).

true

/* as tom is parent of liz, it must be that liz is offspring of tom so it results true */

/* finding all pairs of offspring */

?- offspring(Y,X)

Y = bob
X = pam;
Y = bob,
X = tom;
Y = liz,
X = tom,
Y = ann,
X = bob;
Y = pat,
X = bob.
Y = jim,
X = pat.

(2) /* In 2nd rule we define mother */.

mother(X, Y) :- parent(X,Y), female(X).

/* Since X is parent of Y and also X is female so in that case we define a rule stating it to be mother. */

Example query

?- mother(X,Y)

X = pam,
Y = bob;
X = pat,
Y = jim

③ /* In 3rd rule we define father */

father(X,Y) :- parent(X,Y), male(X).

/* If X is parent of Y and X is male we state of being father. */

Example query:

?- father(X,Y).

X = tom;

Y = bob;

X = tom;

Y = liz;

X = bob,

Y = ann;

X = bob,

Y = pat;

false

/* set of all relation of father having X being father of Y */

/* Note: Here last fact : where pat is parent who is female not male so it returns false as prolong evaluates the last facts */.

④ /* In 4th rule we define grandparent */

grandparent(X,Z) :- parent(X,Y), parent(Y,Z).

/* If Y is parent of Z and X is parent of Y then X is grandparent */

Example query:

?- grandparent(X, Z)

X = pam,
Z = ann;
X = pam,
Z = pat;
X = tom,
Z = ann;
X = tom,
Z = pat;
X = bob,
Z = jim;
false

/* showing all grandparent outputs */.

⑤ In the 5th rule we define grandfather

grandfather(X,Y) :- grandparent(X, Y), male(X).

/* if X is grandparent & also male then our rule state X is grandfather. */

Example query:

?- grandfather(X, Y).

X = tom,
Y = ann;
X = tom,
Y = pat;
X = bob,
Y = jim;
false /* showing all grandfathers */

⑥ In the 6th rule we define sister

sister(x,y) :- parent(z,x), parent(z,y), female(x), not(x = y)

/* here, if z is parent of x as well as y and x ≠ y [not(x = y)] i.e both object is not same and if x is female than it states x is sister of y. */

Example query:

?- sister(x,y).

x = liz,
y = bob;
x = ann,
y = pat;
x = pat,
y = ann;
false

/* finding all sister */

⑦ In the 7th rule we define predecessor

Firstly,

predecessor(x,y) :- parent(x,y).

/* This gives the direct parent */

Then,

predecessor(x,z) :- parent(x,y), predecessor(y,z)

/* This gives indirect predecessor */

In this rule for all X and Z, X is a predecessor of Z if there is a Y such that X is parent of Y and Y is predecessor of Z.

Here, predecessor relation has been defined by two rules (clauses), such relation which is defined by set of clauses is called a procedure.

Example queries:

?- predecessor(X, pat).

X = bob,

X = pam;

X = tom.

false

/* here we are finding predecessor of pat where bob is direct parent (predecessor) and then successive predecessors */

Also

?- predecessor(X, Y).

X = pam,

Y = bob;

X = tom,

Y = bob;

X = tom;

Y = liz;

X = bob

Y = ann;

X = bob,

Y = pat;

X = pat;

y = jim;
x = pam,
y = ann;
x = pam,
y = pat;
x = pam,
y = jim;
x = tom,
y = ann;
x = tom,
y = pat;
x = tom,
y = jim;
x = bob,
y = jim;
False

/* finding all predecessors for instance, bob is predecessor of pam and tom, liz is predecessor of tom, etc