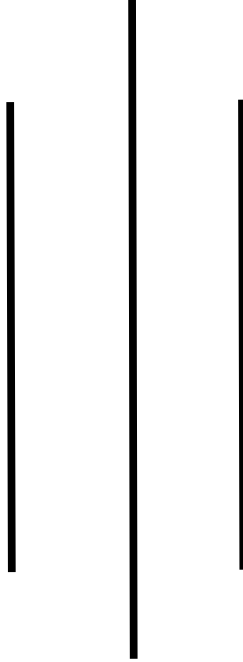


Deerwalk Institute of Technology

Sifal, Kathmandu



Artificial Intelligence Practical - 6

Submitted By:

Name: Sagar Giri

Roll No: 205

Section: A

Submitted To:

Sudan Prajapati

Date: 22 April 2015

Negation as failure:

Example1:

```
enjoys(vincent,X) :- big_kahuna_burger(X),!,fail.
```

```
enjoys(vincent,X) :- burger(X).
```

```
burger(X) :- big_mac(X).
```

```
burger(X) :- big_kahuna_burger(X).
```

```
burger(X) :- whopper(X).
```

```
big_mac(a).
```

```
big_kahuna_burger(b).
```

```
big_mac(c).
```

```
whopper(d).
```

Output:

```
?- enjoys(vincent,a).
```

```
true ;
```

```
false.
```

```
?- enjoys(vincent,b).
```

```
false.
```

```
?- enjoys(vincent,c).
```

```
true ;
```

```
false.
```

```
?- enjoys(vincent,d).
```

```
true.
```

Task1

Mary Likes all animal but snakes. How can we say this in prolog ? It is easy to express one part of this statement .Mary likes any X If X is an animal. But we have to exclude snakes. IF X is a snake then 'Mary likes X' is not true. Otherwise if x is an animal then Mary like X. Make a goal and facts to illustrate above example.

Program Coding:

```
likes(marry,X):-animals(X),!.  
likes(marry,X):-snakes(X),!,fail.
```

```
snakes(cobra).  
snakes(harara).  
snakes(python).
```

```
animals(cat).  
animals(dog).  
animals(snakes).  
animals(elephant).
```

Output:

```
?- likes(marry,cat).  
true.
```

```
?- likes(marry,snake).  
false.
```

```
?- likes(marry,python).  
false.
```

```
?- likes(marry,elephant).  
true.
```

Inequality in prolog

```
?- a=a.  
true.
```

```
?- a=b.  
false.
```

```
?- a=X.  
X = a.
```

```
?- X=a.  
X = a.
```

```
?- X=Y.  
X = Y.
```

```
?- a==a.  
true.
```

```
?- a==b.  
false.
```

```
?- a==X.  
false.
```

```
?- X\=Y.  
false.
```

```
?- X\==Y.  
true.
```

Negation in Prolog:

Fact:

university(uoft).

Output:

```
?- university(uoft).  
true.
```

```
?- not(university(uoft)).  
false.
```

```
?- \+(university(uoft)).  
false.
```

```
?- \+(university(york)).  
true.
```

Negation Implementation:

```
sad(X) :- \+ happy(X).  
happy(X) :- beautiful(X), rich(X).
```

```
rich(bill).  
beautiful(michael).  
rich(michael).  
beautiful(cinderella).
```

Output:

```
?- sad(bill).  
true.
```

```
?- sad(cindrella).  
true.
```

Task2:

facts:

```
color(red). color(orange).color(yellow).
color(green).color( blue).
color(purple).color(black).
color(brown).color(gold).
muddcolor(black).
muddcolor(gold).
awesomescolor(X,Y):-muddcolor(X),muddcolor(Y), X\==Y.
checkInequalaityEarly(X,Y):-X\==Y,muddcolor(X),muddcolor(Y).
notmuddcolorbroken(X):- not(muddcolor(X)).
notmuddcolor(X):-color(X),not(muddcolor(X)).
```

Output:

```
?- muddcolor(X).
X = black ;
X = gold.

?- awesomescolor(X,Y).
X = black,
Y = gold ;
X = gold,
Y = black ;
false.

?- notmuddcolorbroken(X).
false.

?- notmuddcolor(X).
X = red ;
X = orange ;
X = yellow ;
X = green ;
X = blue ;
X = purple ;
X = brown ;
false.
```