Exp 5

To solve a basic programming problem using pyDatalog

PyDatalog adds the logic programming paradigm to python’s toolbox

(a)Encode the following facts and rules in PyDatalog

## Encode the following facts and rules in pyDatalog:

 Bear is big

 Elephant is big

 Cat is small

 Bear is brown

 Cat is black

 Elephant is gray

 An animal is dark if it is black

 An animal is dark if it is brown

Write a query to find which animal is dark and big.

pip install pyDatalog

from pyDatalog import pyDatalog

pyDatalog has inbuilt function to create terms which belong to facts and rules.

pyDatalog.create\_terms('Bear,Elephant,Cat,Black,Gray,Brown,Big,small,size,color,X,Y,dark')

Notify given relation

+size("Bear","Big")

+size("Elephant","Big")

+size("Cat","small")

+color("Bear","Brown")

+color("Cat","Black")

+color("Elephant","Gray")

Declaring relation, print dark of X and size of X should be big

dark(X) <= (color(X,"Black"))

dark(X) <= (color(X,"Brown"))

print(dark(X) &(size(X,'small')))

## 1.

(b)Implement using pyDatalog:

Assume given a set of facts of the form father(name1,name2) (name1 is the father of name2).

a. Define a predicate brother(X,Y) which holds iff X and Y are brothers.

b. Define a predicate cousin(X,Y) which holds iff X and Y are cousins.

c. Define a predicate grandson(X,Y) which holds iff X is a grandson of Y.

d. Define a predicate descendent(X,Y) which holds iff X is a descendent of Y.

e. Consider the following genealogical tree:

## a

## / \

## b c

## / \ |

## d e f

## What are the answers generated by your definitions for the queries:

## brother(X,Y)

## cousin(X,Y)

## grandson(X,Y)

## descendent(X,Y)

from pyDatalog import pyDatalog as py

py.create\_terms("brother,father,cousin,grandson,descendent,X,Y,Z,W,a,b,c,d,e,f")

+father('a','b')

+father('a','c')

+father('b','d')

+father('b','e')

+father('c','f')

brother(X,Y) <= (father(Z,X)) & (father(Z,Y)) & ~(X==Y)

cousin(X,Y) <= (father(Z,X)) & (father(W,Y)) & (brother(Z,W))

grandson(X,Y)<= (father(Y,Z)) & (father(Z,X))

descendent(X,Y) <= (father(Y,X))

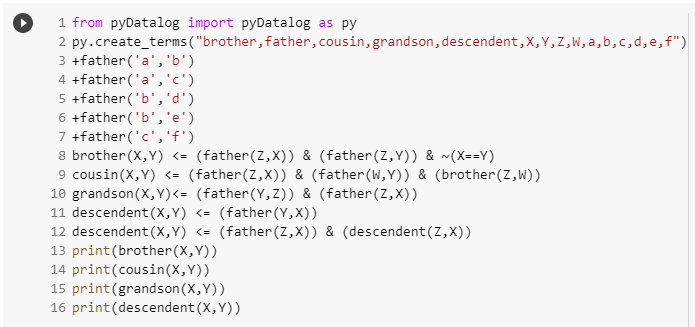
descendent(X,Y) <= (father(Z,X)) & (descendent(Z,X))

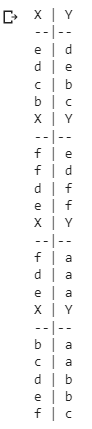
print(brother(X,Y))

print(cousin(X,Y))

print(grandson(X,Y))

print(descendent(X,Y))





https://www.youtube.com/watch?v=odgpKmVD-pQ

https://drive.google.com/drive/folders/1IujA\_O3CXtQcX7m\_9Tui4yoP3AAiI9Dt