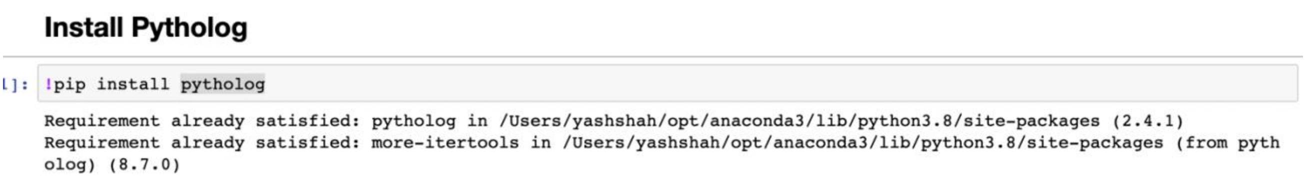
(2022W) COMP-5313-WDE/WB - Artificial Intelligence Computer Science

Exercise 1

First Install the Pytholog using import function. It is a Python library that enables using **logic programming** in python. The aim of the library is to explore ways to use symbolic reasoning with machine learning.

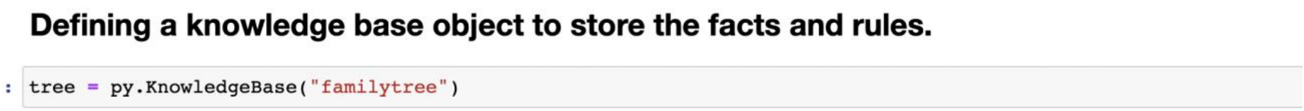


Install Pytholog

```
l]: !pip install pytholog

    Requirement already satisfied: pytholog in /Users/yashshah/opt/anaconda3/lib/python3.8/site-packages (2.4.1)
    Requirement already satisfied: more-itertools in /Users/yashshah/opt/anaconda3/lib/python3.8/site-packages (from pyth
    olog) (8.7.0)
```

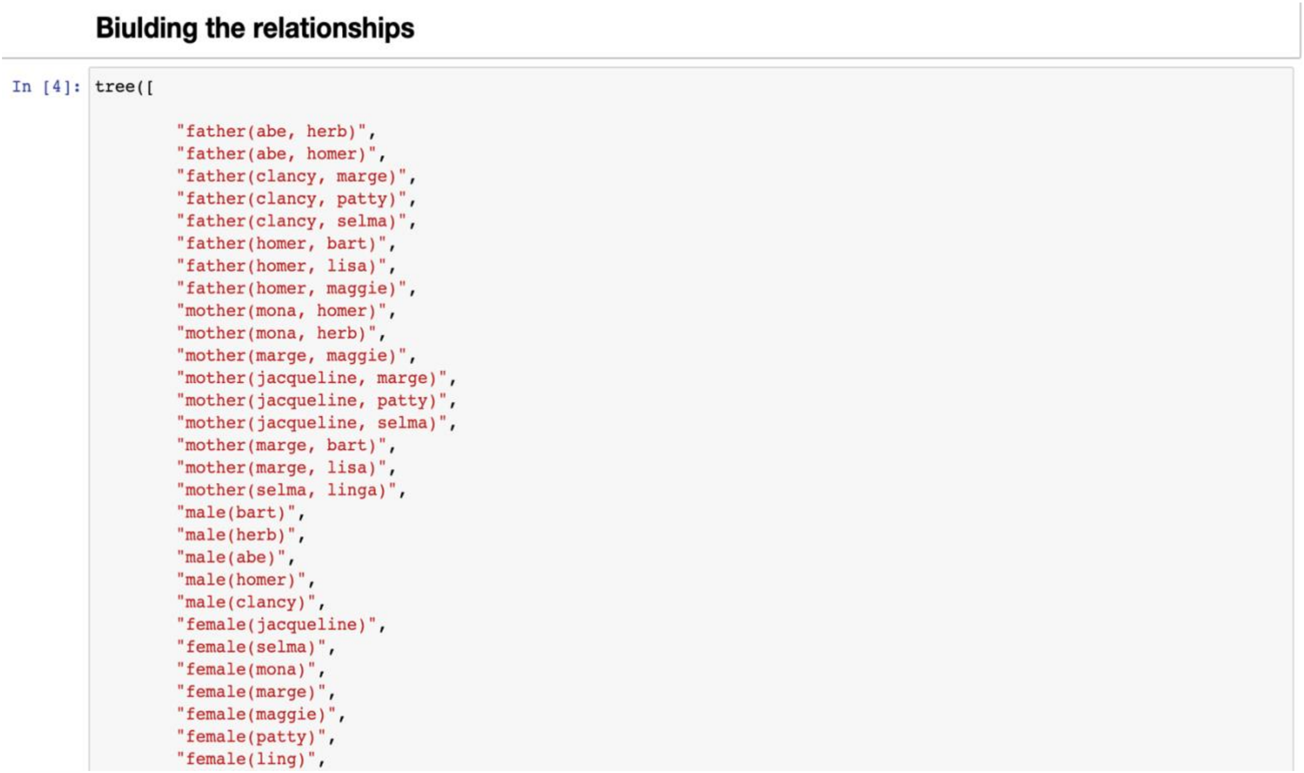
Importing the Pytholog and defining a knowledge based object to store the facts and rules.



Defining a knowledge base object to store the facts and rules.

```
: tree = py.KnowledgeBase("familytree")
```

The next step was to build the relationship

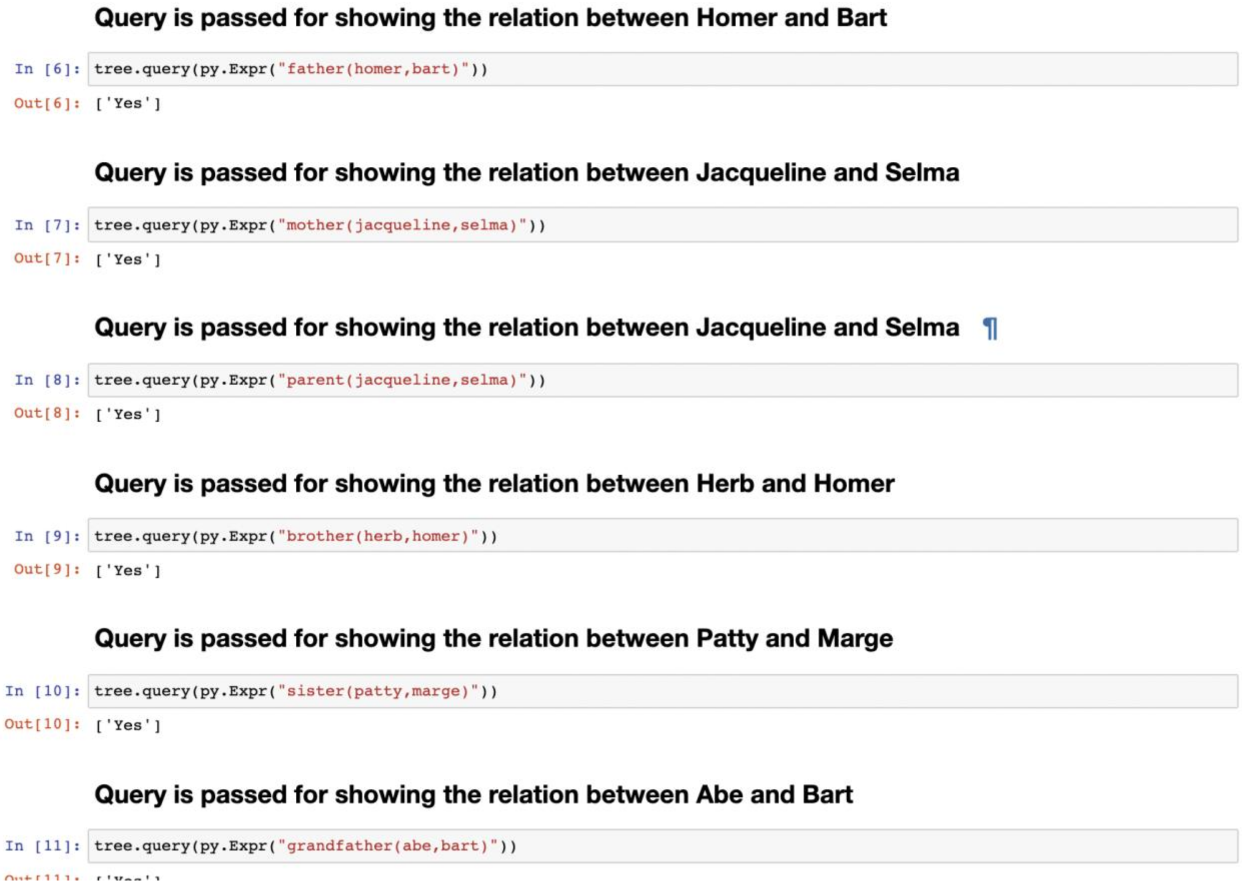


Biulding the relationships

```
In [4]: tree([

                "father(abe, herb)",
                "father(abe, homer)",
                "father(clancy, marge)",
                "father(clancy, patty)",
                "father(clancy, selma)",
                "father(homer, bart)",
                "father(homer, lisa)",
                "father(homer, maggie)",
                "mother(mona, homer)",
                "mother(mona, herb)",
                "mother(marge, maggie)",
                "mother(jacqueline, marge)",
                "mother(jacqueline, patty)",
                "mother(jacqueline, selma)",
                "mother(marge, bart)",
                "mother(marge, lisa)",
                "mother(selma, linga)",
                "male(bart)",
                "male(herb)",
                "male(abe)",
                "male(homer)",
                "male(clancy)",
                "female(jacqueline)",
                "female(selma)",
                "female(mona)",
                "female(marge)",
                "female(maggie)",
                "female(patty)",
                "female(ling)",
```

Passed the query for checking of each relationship



Query is passed for showing the relation between Homer and Bart

```
In [6]: tree.query(py.Expr("father(homer,bart)"))
Out[6]: ['Yes']
```

Query is passed for showing the relation between Jacqueline and Selma

```
In [7]: tree.query(py.Expr("mother(jacqueline,selma)"))
Out[7]: ['Yes']
```

Query is passed for showing the relation between Jacqueline and Selma ¶

```
In [8]: tree.query(py.Expr("parent(jacqueline,selma)"))
Out[8]: ['Yes']
```

Query is passed for showing the relation between Herb and Homer

```
In [9]: tree.query(py.Expr("brother(herb,homer)"))
Out[9]: ['Yes']
```

Query is passed for showing the relation between Patty and Marge

```
In [10]: tree.query(py.Expr("sister(patty,marge)"))
Out[10]: ['Yes']
```

Query is passed for showing the relation between Abe and Bart

```
In [11]: tree.query(py.Expr("grandfather(abe,bart)"))
Out[11]: ['Yes']
```