Neel Zadafiya (1115533)

COMP5313 Artificial Intelligence - Alternative Exercise 1

This document is a part of zip file which contains additional two files which are Neel_Ex1.py and Neel_Ex1.ipynb. Neel_Ex1.py is directly exported from Jupyter netbook and the other file is downloaded as interactive python notebook file.

To repeat the experiment, simply install the pytholog library if its not installed and directly run .ipynb file to get outputs. All the required relations to construct family tree are defined in the program with sufficient comments and all the required new relation ships are defined in the same block. The following relationships has been implemented in the program:

```
# 1. Parent clause : X is parent of Y
# X is either mother of father of Y
 "parent(X,Y) :- father(X,Y)",
 "parent(X,Y) :- mother(X,Y)",

# 2. Brother caluse : X is brother of Y
# X and Y has same father M and same mother N
# X is male and not same as Y
 "brother(X,Y) :- father(M,X), father(M,Y), mother(N,X), mother(N,Y), male(X), neq(X,Y)",

# 3. Sister caluse : X is sister of Y
# X and Y has same father M and same mother N
# X is female and not same as Y
 "sister(X,Y) :- father(M,X), father(M,Y), mother(N,X), mother(N,Y), female(X), neq(X,Y)",

# 6. Grand parent clause : X is grand parent of Y
# X is parent of M and M is parent of Y
 "grand_parent(X,Y) :- parent(X,M), parent(M,Y)",

# 4. Grand father clause: X is grand father of Y
# X is male and grand parent of Y
 "grand_father(X,Y) :- grand_parent(X,Y), male(X)",

# 5. Grand mother clause: X is grand father of Y
# X is female and grand parent of Y
 "grand_mother(X,Y) :- grand_parent(X,Y), female(X)",

# 7. Uncle clause: X is uncle of Y
# Parent M of Y is brother of X
 "uncle(X,Y) :- parent(M,Y), brother(X,M)",
# 8. Aunt clause: X is aunt of Y
# Parent M of Y is sister of X
 "aunt(X,Y) :- parent(M,Y), sister(X,M)",

# 9.1 Nephew clause: X is nephew of Y
# Y is either uncle or aunt of X and X is male
 "nephew(X,Y) :- uncle(Y,X), male(X)",
 "nephew(X,Y) :- aunt(Y,X), male(X)",

# 9.2 Niece clause: X is niece of Y
# Y is either uncle or aunt of X and X is female
 "niece(X,Y) :- uncle(Y,X), female(X)",
 "niece(X,Y) :- aunt(Y,X), female(X)",

# 9.3 Sibling clause: X is sibling of Y
# X is either brother of sister of Y
 "sibling(X,Y) :- brother(X,Y)",
 "sibling(X,Y) :- sister(X,Y)"
```

The numbers are given according to exercise. The program introduces another function to test functionality of knowledge base. The function is test_relation and it takes three arguments. The first argument is relation and other two argument are person.

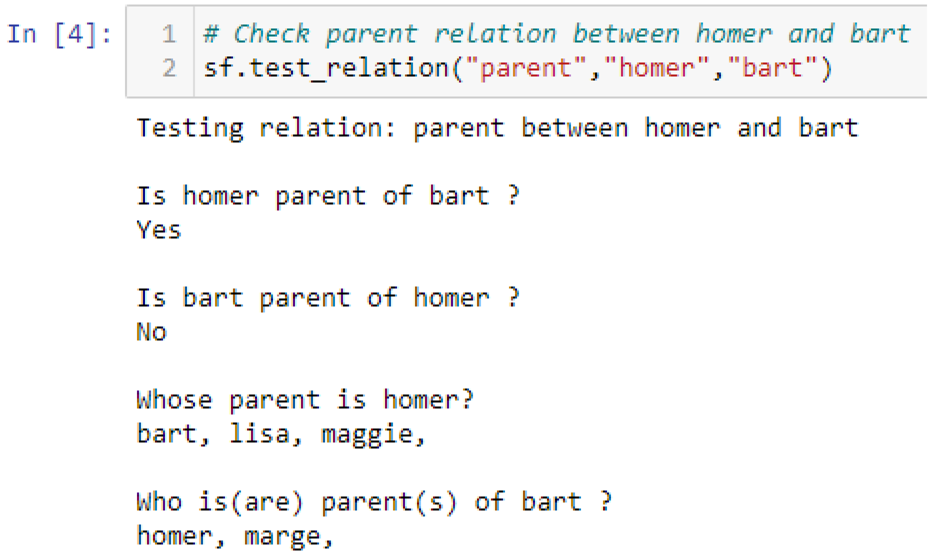
For example, If the relation parent needs to be tested, then function should be used like following:

```
test_relation("parent","homer","bart")
```

Here, parent is relation (clause), homer and bart are people. This function runs four queries. They are following:

1. Parent (homer, bart)
   - This means check if homer is parent of bart. Expected answer is yes.
   - This checks ability of program to check facts.
2. Parent (bart, homer)
   - This means check if bart is parent of homer. Expected answer is no.
   - This checks ability of program to reject false facts.
3. Parent (homer, Who)
   - This means list of people whose parent is homer. Expected answer is bart, lisa, Maggie.
   - This checks ability of program to backtrack.
4. Parent (Who, bart)
   - This means list of people who are parent of bart. Expected answer is homer, marge.
   - This check ability of propgram to backtrack, too.

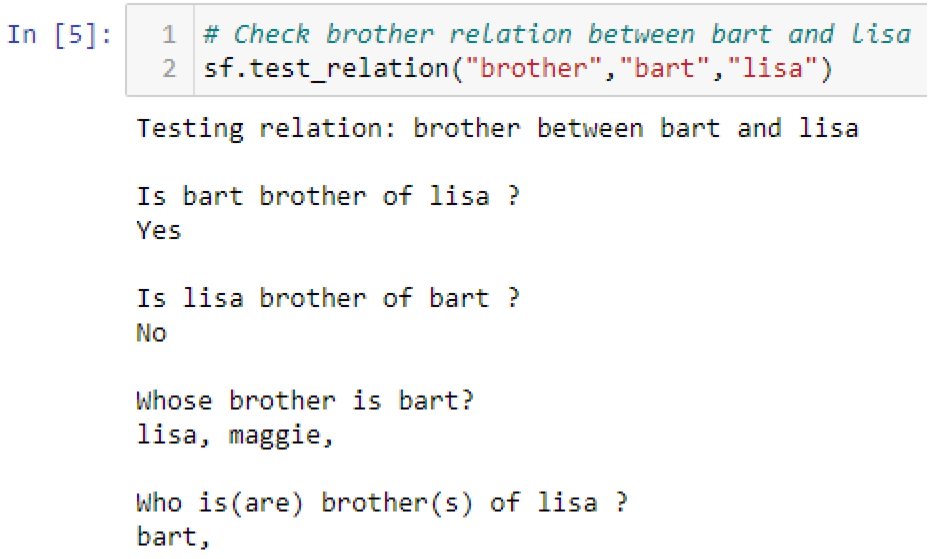
The output in console looks like following:



```
In [4]:
# Check parent relation between homer and bart
sf.test_relation("parent","homer","bart")
```

```
Testing relation: parent between homer and bart

Is homer parent of bart ?
Yes

Is bart parent of homer ?
No

Whose parent is homer?
bart, lisa, maggie,

Who is(are) parent(s) of bart ?
homer, marge,
```

The output is in human readable format so that it can be compared to actual family and it's easy to verify. The output for remaining relations are given below (These are also included in notebook file).



```
In [5]:
# Check brother relation between bart and lisa
sf.test_relation("brother","bart","lisa")
```

```
Testing relation: brother between bart and lisa

Is bart brother of lisa ?
Yes

Is lisa brother of bart ?
No

Whose brother is bart?
lisa, maggie,

Who is(are) brother(s) of lisa ?
bart,
```

In [6]:

```python
# Check sister relation between lisa and bart
sf.test_relation("sister","lisa","bart")
```

```
Testing relation: sister between lisa and bart

Is lisa sister of bart ?
Yes

Is bart sister of lisa ?
No

Whose sister is lisa?
bart, maggie,

Who is(are) sister(s) of bart ?
lisa, maggie,
```

In [7]:

```python
# Check grand father relation between abe and bart
sf.test_relation("grand_father","abe","bart")
```

```
Testing relation: grand_father between abe and bart

Is abe grand_father of bart ?
Yes

Is bart grand_father of abe ?
No

Whose grand_father is abe?
bart, lisa, maggie,

Who is(are) grand_father(s) of bart ?
abe, clancy,
```

In [8]:

```python
# Check grand mother relation between mona and bart
sf.test_relation("grand_mother","mona","bart")
```

```
Testing relation: grand_mother between mona and bart

Is mona grand_mother of bart ?
Yes

Is bart grand_mother of mona ?
No

Whose grand_mother is mona?
bart, lisa, maggie,

Who is(are) grand_mother(s) of bart ?
jacqueline, mona,
```

In [9]:

```python
# Check grand parent relation between abe and bart
sf.test_relation("grand_parent","abe","bart")
```

```
Testing relation: grand_parent between abe and bart

Is abe grand_parent of bart ?
Yes

Is bart grand_parent of abe ?
No

Whose grand_parent is abe?
bart, lisa, maggie,

Who is(are) grand_parent(s) of bart ?
abe, clancy, jacqueline, mona,
```

In [10]:

```
# Check uncle relation between herb and bart
sf.test_relation("uncle","herb","bart")
```

```
Testing relation: uncle between herb and bart

Is herb uncle of bart ?
Yes

Is bart uncle of herb ?
No

Whose uncle is herb?
bart, lisa, maggie,

Who is(are) uncle(s) of bart ?
herb,
```

In [11]:

```
# Check aunt relation between patty and bart
sf.test_relation("aunt","patty","bart")
```

```
Testing relation: aunt between patty and bart

Is patty aunt of bart ?
Yes

Is bart aunt of patty ?
No

Whose aunt is patty?
bart, lisa, maggie, ling,

Who is(are) aunt(s) of bart ?
patty, selma,
```

In [12]:

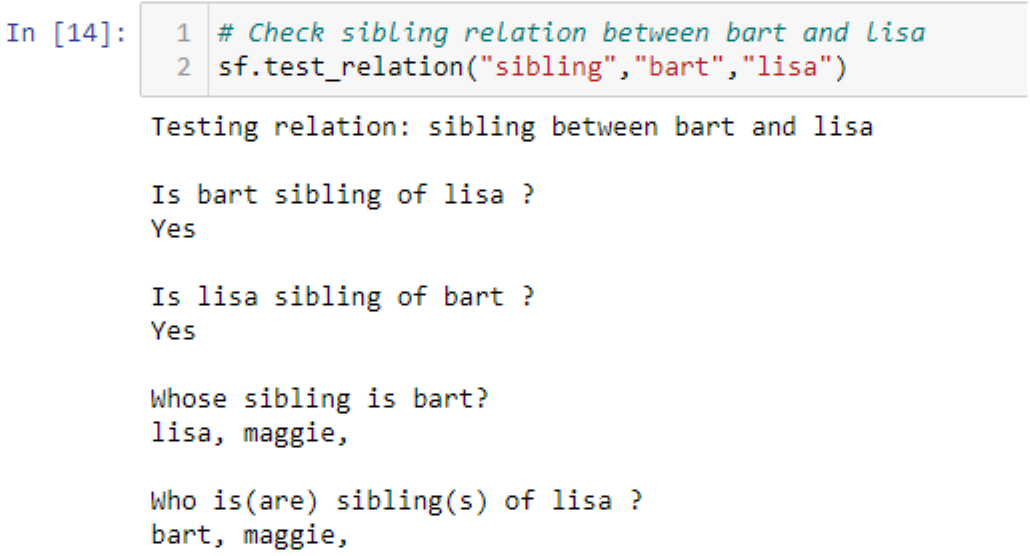
```
# Check nephew relation between bart and herb
sf.test_relation("nephew","bart","herb")
```

```
Testing relation: nephew between bart and herb

Is bart nephew of herb ?
Yes

Is herb nephew of bart ?
No

Whose nephew is bart?
herb, patty, selma,

Who is(are) nephew(s) of herb ?
bart,
```

In [13]:

```
# Check niece relation between lisa and herb
sf.test_relation("niece","lisa","herb")
```

```
Testing relation: niece between lisa and herb

Is lisa niece of herb ?
Yes

Is herb niece of lisa ?
No

Whose niece is lisa?
herb, patty, selma,

Who is(are) niece(s) of herb ?
lisa, maggie,
```



```
# Check sibling relation between bart and lisa
sf.test_relation("sibling","bart","lisa")
```

```
Testing relation: sibling between bart and lisa

Is bart sibling of lisa ?
Yes

Is lisa sibling of bart ?
Yes

Whose sibling is bart?
lisa, maggie,

Who is(are) sibling(s) of lisa ?
bart, maggie,
```

End of ReadMe document.