Ahsanullah University of Science and Technology

Department of Computer Science And Engineering

Course No: 4108

Course Title: Artificial Intelligence Lab

Assignment No: 01

Session: Spring-20

Date of Submission: 31st Jan, 2021

Submitted To:

Tonmoy Hossain Lecturer, Department Of CSE

Rayhan Ahmed Lecturer, Department Of CSE

Submitted by:

Name: Faiza Anan Noor
Year: 4th
Semester: 1st
Lab Group: B2
ID: 17.01.04.093

Lab Exercises :

Questions :

1. Modify the Python and Prolog codes demonstrated above to find the grandparents of somebody.

2. Enrich the KB demonstrated above with ‘brother’, ‘sister’, ‘uncle’ and ‘aunt’ rules in Python and Prolog.

Ans to Question no. 1 :

Code In Prolog:

```
parent('Hasib' , 'Rakib').
parent('Rakib' , 'Sohel').
parent('Rakib' , 'Rebeka').
parent('Rakib' , 'Russel').
parent('Sohel' , 'Kathy').
parent('Rashid', 'Hasib').
parent('Rashid', 'Karim').
parent('Shamsun' , 'Murad').
parent('Murad', 'Zunaira').
parent('Zunaira', 'Zuhair').
```

```
grandparent(X, Z):- parent(X, Y), parent(Y, Z).
findGp :- write(' Grandchildren: '),  read(X),  write('Grandparent: '),
                    grandparent(Gp, X),  write(Gp),  tab(5),fail.
findGp.
```

Explanation :

1. **parent('Rashid' , 'Karim'):**

In our prolog code, the term parent('Rashid' , 'Karim') refers to a fact what means that Rashid is a parent of Karim. Parent is a predicate in this case. Let's consider these 3 examples same of the same manner.

```
parent('Sohel' , 'Kathy').
parent('Rashid', 'Hasib').
parent('Rashid', 'Karim').
```

Here all these 3 things denote that the parameter on the left is the parent of the parameter on the right.

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

This is a rule which notes that X will be the grandparent of Z if X is the parent of Y and Y is the parent of Z.

If Z is the grandchild than his father Y will be the son or daughter of X.

For example,

```
parent('Shamsun' , 'Murad').
parent('Murad', 'Zunaira').
```

For this case, Shamsun is the parent of Murad and Murad is the parent of Zunaira. So in this case Shamsun will be the grandparent of Zunaira.

3. **findGp :- write(' Grandchildren: '), read(X), write('Grandparent: '), grandparent(Gp, X), write(Gp), tab(5),fail.**

Here let’s analyze each of these terms.

Here findGp is a rule which will give us the name of of Grandparent.

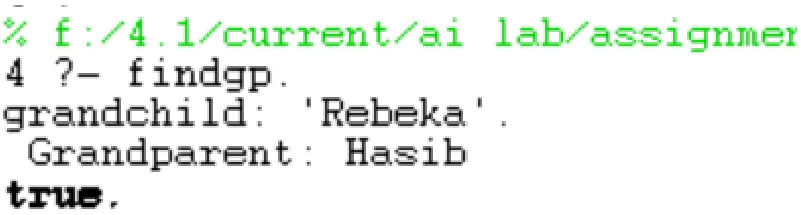
Write() will be used to write something on the console.

Here read() will take the name of GrandChildren as input and store it in a variable called ‘X’. And ‘Gp’ will figure out the name of the Grandparent for our grandparent ‘X’. and write(Gp) will print that on the console. By passing in Gp, X variables in grandparent, we can get our grandparent and print it out.

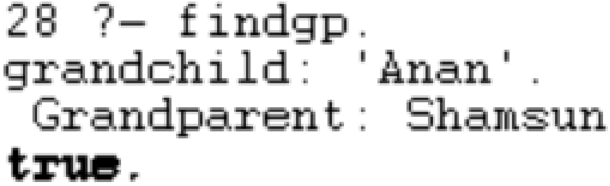
Here tab(5) gives us tabs of space .

Here Fail does the work of making the rule false and backtrack again from the last found tag and the final findGp always makes the rule true.

Prolog Output :



```
% f:/4.1/current/ai lab/assignmer
4 ?- findgp.
grandchild: 'Rebeka'.
 Grandparent: Hasib
true.
```



```
28 ?- findgp.
grandchild: 'Anan'.
 Grandparent: Shamsun
true.
```

Here If we call findgp,

Then we’re asked to input the grandchild name. In our case we gave Rebeka as input. If we do so we will get the appropriate grandparent as output for Rebeka which is Hasib. We did the same for the other example as well where we got the correct grandparent as an output for our given input.

Code In Python :

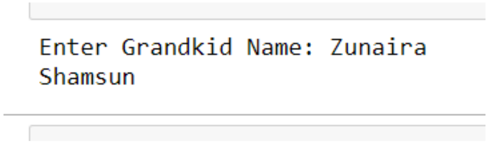
```
tupleList1 = [('parent', 'Happy', 'Faiza'), ('parent', 'Happy', 'Zahin'),
            ('parent', 'Faiza', 'Shiro'), ('parent', 'Shiro', 'Shiro2.0'),
            ('parent', 'Shamsun', 'Happy'), ('parent', 'Shamsun', 'Murad'),
            ('parent', 'Murad', 'Zunaira'), ('parent', 'Murad', 'Zuhair') ]
# Finding Grandparent of someone
X = str(input('Enter Grandkid Name'))


for i in range(8):
    if ((tupleList1[i][0] == 'parent')):
        for j in range(8):
            if ((tupleList1[j][0] == 'parent') & (tupleList1[i][2] == tupleList1[j][1]) &
(tupleList1[j][2] == X)):
                print(tupleList1[i][1])
```

Explanation :

Here, we take the input of whose grandparent we want to find. Then we check and look for the parent of the parent of our input in this way:

```
for i in range(8):
    if ((tupleList1[i][0] == 'parent')):
        for j in range(8):
            if ((tupleList1[j][0] == 'parent') & (tupleList1[i][2] == tupleList1[j][1]) &
(tupleList1[j][2] == X)):
                print(tupleList1[i][1])
```

Python Output :



```
Enter Grandkid Name: Zunaira
Shamsun
```

Here in this output we can see that ,we gave the name of the name of the grandchild as input which we gave ‘Zunaira’. So then we got the grandparent name which is ‘Shamsun’ which is true for our case.

Ans to Question no. 2 :

Code In Prolog:

```
parent('Faiza','Anan').

parent('Shariba','Shababa').

parent('Shamsun','Happy').

parent('Shamsun','Tamanna').

parent('Murad','Zunaira').

parent('Shamsun','Murad').

parent('Happy','Anan').

parent('Murad','Zuhair').

parent('Faiza','Shiro').

parent('Happy','Zahin').

parent('Hasib' , 'Rakib').
parent('Rakib' , 'Sohel').
parent('Rakib' , 'Rebeka').
parent('Rakib' , 'Russel').

female('Faiza').
female('Tamanna').
female('Anan').
female('Noor').
female('Zahin').
female('Zunaira').
female('Happy').

female('Shariba').
female('Shababa').
male('Zuhair').
male('Murad').
male('Sohel').
male('Russel').
male('Rakib').
grandparent(X,Z):-parent(X,Y),parent(Y,Z).
brothers(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z),male(X),male(Z).
siblings(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z).
aunts(X,Z):-parent(Y,Z),sisters(X,Y),female(X),female(Y).
```

```
uncle(X,Z):-parent(Y,Z),male(X),siblings(Y,X).

sisters(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z),female(X),female(Z).


findbrother:- write(' Name:'), read(Z),write('Brother:
'),brothers(X,Z),write(X),tab(5), fail.

findbrother.

findsister :- write(' Name:'), read(Z), write('Sister:'), sisters(X,Z),write(X),tab(5),
fail.

findsister.

finduncle :- write('Name: '), read(Z),write('Uncle: '),uncle(X,Z),write(X),tab(5),fail.

finduncle.

findaunt:- write(' Name:'), read(Z), write('Aunt:'),aunts(X,Z),write(X),tab(5), fail.

findaunt.
```

Explanation :

For finding brother of someone :

```
brothers(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z),male(X),male(Z).
```

This rule denotes that if Y is a parent of X ,Z and if X,Z are not the same person and if X is a male the X and Z are brothers.

findbrother:- write(' Name:'), read(Z),write('Brother: '),brothers(X,Z),write(X),tab(5), fail.

findbrother.

This takes the input of whose brother we want to find and then it prints out the brother’s name on the console.

For finding sister of someone :

sisters(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z),female(X),female(Z).

This rule denotes that if Y is a parent of X ,Z and if X,Z are not the same person and if X is a female the X and Z are brothers.

findsister :- write(' Name:'), read(Z), write('Sister:'), sisters(X,Z),write(X),tab(5), fail.

findsister.

This takes the input of whose sister we want to find and then it prints out the sister’s name on the console.

For finding uncle of someone :

uncle(X,Z):-brother(X,Y),parent(Y,Z),male(X).

In this rule for uncle X,Y should be brother and Y should be parent of Z and X should be male.

siblings(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z).

finduncle :- write('Name: '), read(Z),write('Uncle: '),uncle(X,Z),write(X),tab(5),fail.

finduncle.

Here we will take input of whose uncle we want to find in Z. And then we’ll find out the parent of Z, and the siblings of the parent of Z. And we are also checking if that person is male or not. Then finally we are printing the uncles name on the console.

For finding aunt of someone :

siblings(X,Z):-parent(Y,X),parent(Y,Z),not(X=Z).

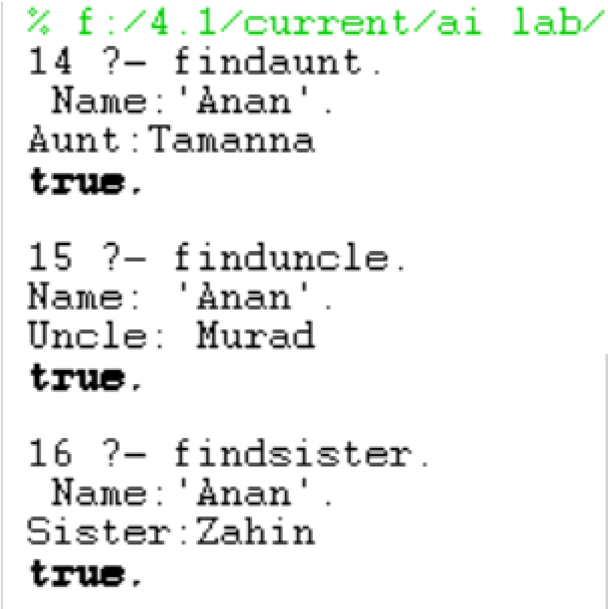
aunts(X,Z):-parent(Y,Z),sisters(X,Y),female(X),female(Y).

findaunt:- write(' Name:'), read(Z), write('Aunt:'),aunts(X,Z),write(X),tab(5), fail.

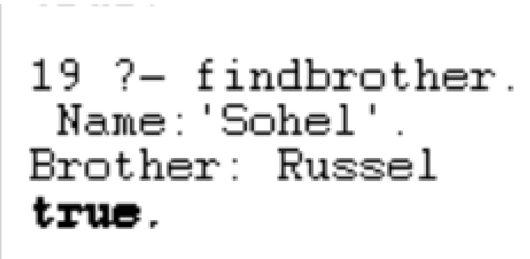
findaunt.

Here we will take input of whose aunt we want to find in Z. And then we'll find out the parent of Z, and the siblings of the parent of Z. And we are also checking if that person is female or not. Then finally we are printing the aunts name on the console.

Prolog Output :



```
% f:/4.1/current/ai lab/
14 ?- findaunt.
 Name:'Anan'.
Aunt:Tamanna
true.

15 ?- finduncle.
Name: 'Anan'.
Uncle: Murad
true.

16 ?- findsister.
 Name:'Anan'.
Sister:Zahin
true.
```



```
19 ?- findbrother.
 Name:'Sohel'.
Brother: Russel
true.
```

- For brother we call findbrother and give a name who must be a male and the output will show us if that person is someones brother or not.
- For sister we call findsister and give a name who must be a male and the output will show us if that person is someones sister or not.

- For uncle we call finduncle and have to give a name and the output will show us the name of nephew or niece.
- For aunty we call findaunt and have to give a name and the output will show us the name of nephew or niece.

Code in Python:

```
tupleList1 = [('parent', 'Happy', 'Faiza'), ('parent', 'Happy', 'Zahin'),
            ('parent', 'Faiza', 'Shiro'), ('parent', 'Shiro', 'Shiro2.0'),
            ('parent', 'Shamsun', 'Happy'), ('parent', 'Shamsun', 'Murad'),
            ('parent', 'Murad', 'Zunaira'), ('parent', 'Murad', 'Zuhair'), ]



tupleList2 = [('male', 'Murad'), ('female', 'Faiza'), ('male', 'Murad'),
            ('female', 'Faiza'), ('female', 'Happy'), ('female', 'Shamsun'),
            ('male', 'Shiro2.0'), ('male', 'Zunaira'), ('male', 'Zuhair'),
            ('female', 'Zahin'), ('male', 'Murad'), ('male', 'Murad'), ]
# Finding sisters


X = str(input("Enter name to find sisters of "))


i = 0
flag = 0


for i in range(8):
    if ((tupleList1[i][2] == X) & (tupleList1[i][0] == 'parent')):
        for j in range(8):
            flag = 0
            if ((tupleList1[j][1] == tupleList1[i][1]) & (i != j)):
                for k in range(12):
                    if ((tupleList1[j][2] == tupleList2[k][1]) & (tupleList2[k][0] == 'female')):
                        flag = 1
                        sibling = tupleList1[j][2]
                        break

            if (flag == 1):
                print(sibling)

X = str(input("Enter name to find brothers of "))
#finding brothers
for i in range(8):
    if ((tupleList1[i][2] == X) & (tupleList1[i][0] == 'parent')):
        for j in range(8):
            flag = 0
            if ((tupleList1[j][1] == tupleList1[i][1]) & (i != j)):
                for k in range(12):
                    if ((tupleList1[j][2] == tupleList2[k][1]) & (tupleList2[k][0] == 'male')):
                        flag = 1
                        sibling = tupleList1[j][2]
                        break
            if (flag == 1):
                print(sibling)
X = str(input("Enter name to find aunts of "))

for i in range(8):

    if((tupleList1[i][0]=='parent')&(tupleList1[i][2]==X)):
        for j in range(8):
            if((tupleList1[j][2]==tupleList1[i][1])&(i!=j)):
                grandParent=tupleList1[j][1]
                parent=tupleList1[j][2]
                break

        for k in range(8):

            if((tupleList1[k][1]==grandParent)&(tupleList1[k][2]!=parent)):
               # print((tupleList1[k][2]))
                for  l in range(12):
                    flag = 0
                    aunt=""
                    if((tupleList2[l][1]==tupleList1[k][2])&(tupleList2[l][0]=='female')):

                        flag=1
            aunt=tupleList1[k][2]
            break

          else:
            flag=0

        if(flag==1):
          print(aunt)

X = str(input("Enter name to find uncles of "))

for i in range(8):

  if((tupleList1[i][0]=='parent')&(tupleList1[i][2]==X)):
    for j in range(8):
      if((tupleList1[j][2]==tupleList1[i][1])&(i!=j)):
        grandParent=tupleList1[j][1]
        parent=tupleList1[j][2]
        break

    for k in range(8):

      if((tupleList1[k][1]==grandParent)&(tupleList1[k][2]!=parent)):
       # print((tupleList1[k][2]))
```

```
                for  l in range(12):
                    flag = 0
                    aunt=""
                    if((tupleList2[l][1]==tupleList1[k][2])&(tupleList2[l][0]=='male')):

                        flag=1
                        aunt=tupleList1[k][2]
                        break

                    else:
                        flag=0

                if(flag==1):
                    print(aunt)
```

Explanation:

Code snippet for finding brother :

```
for i in range(8):
    if ((tupleList1[i][2] == X) & (tupleList1[i][0] == 'parent')):
        for j in range(8):
            flag = 0
            if ((tupleList1[j][1] == tupleList1[i][1]) & (i != j)):
                for k in range(12):
                    if ((tupleList1[j][2] == tupleList2[k][1]) & (tupleList2[k][0] == 'male')):
                        flag = 1
                        sibling = tupleList1[j][2]
```

Here firstly we take input of chose brother we want to find. Then we check to see whose parent matches with the person whose name we input. We also check if that person is a male from our TupleList for differentiating between male and females

Code snippet for finding sister :

```
for i in range(8):
    if ((tupleList1[i][2] == X) & (tupleList1[i][0] == 'parent')):
        for j in range(8):
            flag = 0
            if ((tupleList1[j][1] == tupleList1[i][1]) & (i != j)):
                for k in range(12):
                    if ((tupleList1[j][2] == tupleList2[k][1]) & (tupleList2[k][0] == 'female')):
                        flag = 1
                        sibling = tupleList1[j][2]
```

Here firstly we take input of chose sister we want to find. Then we check to see whose parent matches with the person whose name we input. We also check if that person is a female from our TupleList for differentiating between male and females

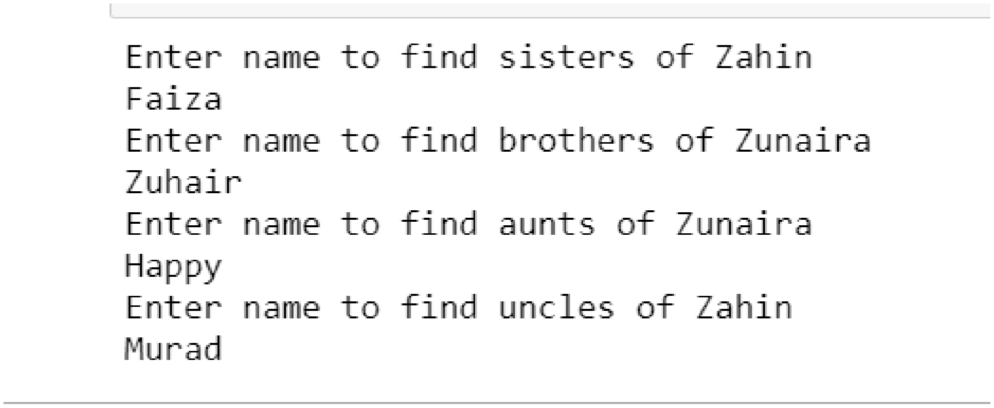
Logic for finding uncle :

For this case first we input the person whose uncle we want to find. Then we find the grandparent and parent of our input. Then we check to see who shares the same parent as the parent of our input. That is the grandparent we found earlier should be the parent to both our uncle, and also the parent of our input. And we have to check whether our output is male or not using the TupleList to differentiate between male and females.

Logic for finding aunt :

For this case first we input the person whose uncle we want to find. Then we find the grandparent and parent of our input. Then we check to see who shares the same parent as the parent of our input. That is the grandparent we found earlier should be the parent to both our uncle, and also the parent of our input. And we have to check whether our output is female or not using the TupleList to differentiate between male and females.

Python Output :



```
Enter name to find sisters of Zahin
Faiza
Enter name to find brothers of Zunaira
Zuhair
Enter name to find aunts of Zunaira
Happy
Enter name to find uncles of Zahin
Murad
```

- To find the sister of someone, we gave out input which is “Zahin” and it gave us the appropriate result which is “Faiza”.
- To find the brother of someone, we gave out input which is “Zunaira” and it gave us the appropriate result which is “Zuhair”.
- To find the aunt of someone, we gave out input which is “Zunaira” and it gave us the appropriate result which is “Happy”.
- To find the uncle of someone, we gave out input which is “Zahin” and it gave us the appropriate result which is “Murad”.