School of Data and Computer Science, Sun Yat-sen University

Lab 5 Family Problem (Prolog)

17341015 Hongzheng Chen

September 26, 2019

Contents

1 About Cousin and Removed

What Is a First Cousin, Twice Removed?

If someone walked up to you and said, "Howdy, I'm your third cousin, twice removed," would you have any idea what they meant? Most people have a good understanding of basic relationship words such as "mother," "father," "aunt," "uncle," "brother," and "sister." But what about the relationship terms that we don't use in everyday speech? Terms like "second cousin" and "first cousin, once removed"? We don't tend to speak about our relationships in such exact terms ("cousin" seems good enough when you are introducing one person to another), so most of us aren't familiar with what these words mean.

Relationship Terms

Sometimes, especially when working on your family history, it's handy to know how to describe your family relationships more exactly. The definitions below should help you out.

Cousin (a.k.a "first cousin")

Your first cousins are the people in your family who have two of the same grandparents as you. In other words, they are the children of your aunts and uncles.

Second Cousin

Your second cousins are the people in your family who have the same great-grandparents as you., but not the same grandparents.

Third, Fourth, and Fifth Cousins

Your third cousins have the same great great grandparents, fourth cousins have the same great-great-great-grandparents, and so on.

Removed

When the word "removed" is used to describe a relationship, it indicates that the two people are from different generations. You and your first cousins are in the same generation (two generations younger than your grandparents), so the word "removed" is not used to describe your relationship.

The words "**once removed**" mean that there is a difference of one generation. For example, your mother's first cousin is your first cousin, once removed. This is because your mother's first cousin is one generation younger than your grandparents and you are two generations younger than your grandparents. This one-generation difference equals "once removed."

Twice removed means that there is a two-generation difference. You are two generations younger than a first cousin of your grandmother, so you and your grandmother's first cousin are first cousins, twice removed.

2 Problem Description

Please fulfill the following tasks by using `Prolog`:

1. Write sentences describing the predicates **Grandchild**, **Greatgrandparent**, **Ancestor**, **Brother**, **Sister**, **Daughter**, **Son**, **FirstCousin**, **BrotherInLaw**, **SisterInLaw**, **Aunt**, and **Uncle**. *Hint: you can define these predicates by choosing child, sibling, male, female, father, mother, and so on.*

2. Find out the proper definition of ***m*th cousin *n* times removed**, in other words, define the predicate `mthCousinNremoved(X,Y,M,N)`. *Hint: You'd better define the predicate* `distance(X,Y,N)` *by recursion (please refer to* `hanoi.pl`*) to show there are N generations between X and Y in advance.*

3. Write down the basic facts depicted in the family tree in Figure 2.

4. ASK it who are **Elizabeth's grandchildren**, **Diana's brothers-in-law**, **Zara's great-grandparents**, and **Eugenie's ancestors**.

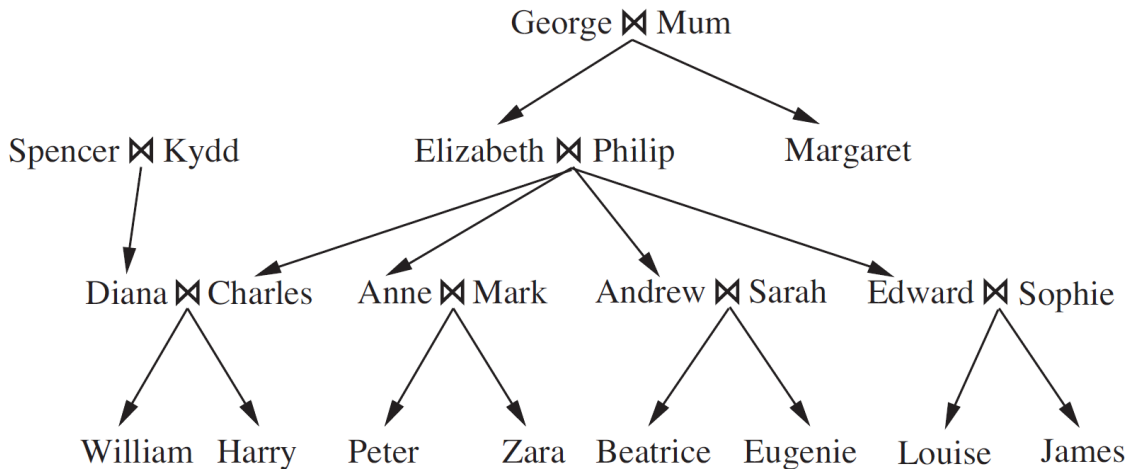

Figure 1: A typical family tree. The symbol ⋈ connects spouses and arrows point to children.

Figure 2: A typical family tree (Chinese version).

3 Tasks

1. Please complete the `Prolog` codes. There are several tutorials in the folder and I will explain the usage of Prolog in class.

2. Write the related codes and take a screenshot of the running results in the file named E05_YourNumber.pdf, and send it to ai_201901@foxmail.com.

4 Codes

The Prolog code is very intuitive and reads as what it declares, but there are several things to mention.

- Be careful of the possible values of `X` and `Y`, since some rules may be satisfied when `X=Y`, which should have been eliminated. For example, `sibling(X,Y)` needs to enforce constraint `X \== Y`, or `X` and `Y` may have the same parent, but they are also the same.
- `firstCousin(X,Y)` not only needs to guarantee the grandparent of `X` and `Y` are the same, but also needs to make sure they do not have the same parent.
- The hardest part of this experiment is the design of the cousin logic. I firstly design an axillary function `mthAncestor(X,Y,M)` for giving out the m-th ancestor `Y` of `X`. Notice the foundation part that `M` is equal to zero, and it should return an identity function.
- Then I implement the `mthCousin(X,Y,M)`. Many people only make `X` and `Y` have the same (m+1)-th ancestor `Z` (the provided `distance` function), which is not correct. Based on the definition of the m-th cousin, we should make sure `X` and `Y` get the same ancestor following different paths. Thus, the child of `Z` on `X`'s path and the child of `Z` on `Y`'s path should not be the same. So we have the constraints in Line 122.
- Finally is the `mthCousinNremoved(X,Y,M,N)` rule which is similar to the implementation of the rule `mthCousin`. The same ancestor `Z` should have different children for `X` and `Y` to get up to there, as constrained in Line 126.

```
1  /* facts */
2  oneself(george,george).
3  oneself(mum,mum).
4  oneself(spencer,spencer).
5  oneself(kydd,kydd).
6  oneself(elizabeth,elizabeth).
7  oneself(philip,philip).
8  oneself(margaret,margaret).
9  oneself(diana,diana).
10 oneself(charles,charles).
11 oneself(anne,anne).
12 oneself(mark,mark).
13 oneself(andrew,andrew).
14 oneself(sarah,sarah).
15 oneself(edward,edward).
16 oneself(sophie,sophie).
17 oneself(william,william).
18 oneself(harry,harry).
19 oneself(peter,peter).
20 oneself(zara,zara).
21 oneself(beatrice,beatrice).
22 oneself(eugenie,eugenie).
23 oneself(louise,louise).
24 oneself(james,james).
25
26 %% sex
27 % #1 gen
28 male(george).
29 female(mum).
```

```
% #2 gen
female(elizabeth).
male(philip).
female(margaret).
male(spencer).
female(kydd).
% #3 gen
female(diana).
male(charles).
female(anne).
male(mark).
male(andrew).
female(sarah).
male(edward).
female(sophie).
% #4 gen
male(william).
male(harry).
male(peter).
female(zara).
female(beatrice).
female(eugenie).
female(louise).
male(james).

%% parent
parent(diana,spencer).
parent(diana,kydd).
parent(william,diana).
parent(william,charles).
parent(harry,diana).
parent(harry,charles).
parent(charles,elizabeth).
parent(charles,philip).
parent(anne,elizabeth).
parent(anne,philip).
parent(peter,anne).
parent(peter,mark).
parent(zara,anne).
parent(zara,mark).
parent(andrew,elizabeth).
parent(andrew,philip).
parent(beatrice,andrew).
parent(beatrice,sarah).
parent(eugenie,andrew).
parent(eugenie,sarah).
parent(louise,edward).
parent(louise,sophie).
parent(james,edward).
parent(james,sophie).
parent(edward,elizabeth).
parent(edward,philip).
parent(elizabeth,george).
parent(elizabeth,mum).
parent(margaret,george).
parent(margaret,mum).
```

```
86
87  /* rules */
88  child(X,Y) :- parent(Y,X).
89  father(X,Y) :- parent(X,Y), male(Y).
90  mother(X,Y) :- parent(X,Y), female(Y).
91  husband(X,Y) :- female(X), male(Y), father(Z,Y), mother(Z,X).
92  wife(X,Y) :- male(X), female(Y), father(Z,X), mother(Z,Y).
93  spouse(X,Y) :- husband(X,Y) ; wife(X,Y).
94  son(X,Y) :- child(X,Y), male(Y).
95  daughter(X,Y) :- child(X,Y), female(Y).
96  sibling(X,Y) :- parent(X,Z), parent(Y,Z), X \== Y.
97  brother(X,Y) :- sibling(X,Y), male(Y).
98  sister(X,Y) :- sibling(X,Y), female(Y).
99  grandfather(X,Y) :- parent(X,Z), father(Z,Y).
100 grandmother(X,Y) :- parent(X,Z), mother(Z,Y).
101 grandchild(X,Y) :- child(X,Z), child(Z,Y).
102 grandparent(X,Y) :- parent(X,Z), parent(Z,Y).
103 greatGrandparent(X,Y) :- grandparent(X,Z), parent(Z,Y).
104 ancestor(X,Y) :- parent(X,Y). % Base
105 ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y). % Recursion
106 aunt(X,Y) :- grandparent(X,Z), parent(Y,Z), \+(mother(X,Y)), female(Y).
107 uncle(X,Y) :- grandparent(X,Z), parent(Y,Z), \+(father(X,Y)), male(Y).
108 %% the husband of your sister or brother
109 %% or the brother of your husband or wife
110 %% or the man who is married to the sister
111 %% or brother of your wife or husband
112 brotherInLaw(X,Y) :- (spouse(X,Z), brother(Z,Y)) ; (sister(X,Z), husband(Z,Y)).
113 sisterInLaw(X,Y) :- (spouse(X,Z), sister(Z,Y)) ; (brother(X,Z), wife(Z,Y)).
114 % cousins
115 firstCousin(X,Y) :- grandparent(X,Z), grandparent(Y,Z), X \== Y, \+(sibling(X,Y)).
116 mthAncestor(X,Y,0) :- oneself(X,Y).
117 mthAncestor(X,Y,1) :- parent(X,Y).
118 mthAncestor(X,Y,M) :- parent(X,Z), M1 is M-1, mthAncestor(Z,Y,M1).
119 mthCousin(X,Y,1) :- firstCousin(X,Y).
120 mthCousin(X,Y,M) :-
121     M1 is M+1, mthAncestor(X,Z,M1), mthAncestor(Y,Z,M1), X \== Y,
122     mthAncestor(X,A,M), mthAncestor(Y,B,M), A \= B, parent(A,Z), parent(B,Z).
123     %% mthAncestor(X,A,M), mthAncestor(Y,B,M), A \= B, \+(spouse(A,B)).
124 mthCousinNremoved(X,Y,M,N) :-
125     M1 is M+1, mthAncestor(X,Z,M1), M2 is M-N+1, mthAncestor(Y,Z,M2), X \== Y,
126     mthAncestor(X,A,M), M3 is M-N, mthAncestor(Y,B,M3), parent(A,Z), parent(B,Z), A \== B.
```

5 Results

The results are shown below, where I test all the cases in the questions.

```
chhzh123@DESKTOP-PV2UBJL: /mnt/d/Assignments/ArtificalIntelligence/E05_Family
chhzh123@DESKTOP-PV2UBJL:/mnt/d/Assignments/ArtificalIntelligence/E05_Family$ prolog
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [family].
true.

?- findall(X,grandchild(elizabeth,X),Z).
Z = [william, harry, peter, zara, beatrice, eugenie, louise, james].

?- findall(X,grandparent(william,X),Z).
Z = [spencer, kydd, elizabeth, philip].

?- findall(X,ancestor(peter,X),Z).
Z = [anne, mark, elizabeth, philip, george, mum].

?- brother(harry,X).
X = william .

?- sister(beatrice,X).
X = eugenie .

?- findall(X,daughter(andrew,X),Z).
Z = [beatrice, eugenie].

?- findall(X,son(philip,X),Z).
Z = [charles, andrew, edward].

?- firstCousin(william,X).
X = peter .

?- botherInLaw(andrew,X).
Correct to: "brotherInLaw(andrew,X)"? yes
X = mark
Unknown action: e (h for help)
Action?
Unknown action: s (h for help)
Action? .

?- sisterInLaw(anne,X).
X = diana .
```

```
chhzh123@DESKTOP-PV2UBJL: /mnt/d/Assignments/ArtificalIntelligence/E05_Family
?- aunt(william,X).
X = anne .

?- uncle(william,X).
X = andrew .

?- mthCousinNremoved(peter,X,1,0).
X = william .

?- mthCousinNremoved(peter,X,1,1).
X = charles .

?- mthCousinNremoved(peter,X,2,2).
X = margaret .

?- mthCousinNremoved(peter,X,2,1).
false.

?-
```

```
chhzh123@DESKTOP-PV2UBJL: /mnt/d/Assignments/ArtificalIntelligence/E05_Family

?- findall(X,grandchild(elizabeths,X),Z).
Z = [].

?- findall(X,grandchild(elizabeth,X),Z).
Z = [william, harry, peter, zara, beatrice, eugenie, louise, james].

?- findall(X,brotherInLaw(diana,X),Z).
Z = [andrew, edward, andrew, edward, andrew, edward, andrew, edward].

?- findall(X,greatGrandparent(zara,X),Z).
Z = [george, mum].

?- findall(X,ancestor(eugenie,X),Z).
Z = [andrew, sarah, elizabeth, philip, george, mum].

?-
```